Methodology of Programming and Programming in C

(Common for BCA & BSc Comp.Sc. under Off Campus Stream)

1) Character set of C language contains ?

1. Alphabets
2. Digits
3. Special Symbols
4. All of these

2) A Variable name can have ?

1. Any special symbol
2. blank spaces
3. double
4. char

3) In C language , one of the following is not a valid data type :

1. long
2. float
3. double
4. char

4) The format string %lf is used for ?

1. float
2. double
3. unsigned int
4. long double

5) A variable of type unsigned int can have a value in the range ?

1. -32768 to +32767
2. 0 to 32767
3. 0 to 65535
4. -32767 to +32767

6) Which data type is not a primary data type ?

1. int
2. array
3. float
4. char

7) Which of the format string is not valid ?

1. %ld
2. %lf
3. %lu

4. %lc

8) Which is the valid string data ?

1. 'A'
2. A
3. "A"
4. none of these

9) How much memory is required to store a value of type double ?

1. 4 bytes
2. 6 bytes
3. 8 bytes
4. 10 bytes

10) The modifier which is used to declare a variable as constant ?

1. short
2. signed
3. unsigned
4. const
5. **Answers :**

1)	All of these	6)	array
2)	underscore	7)	%lc
3)	long	8)	"A"
4)	double	9)	8 bytes
5)	0 to 65535	10)	const

(11)

```
void main(){
        clrscr();
        printf("%d",sizeof(3.8));
        getch();
}
```

Which of the following is true?
(a)4
(b)8
(c)10
(d)Compiler error
(e)None of these

(12)

```
void main(){
        char *str1="powla";
        char *str2="er";
```

```
        clrscr();
        printf("%s\b\b%s",str1,str2);
        getch();
}
```
Which of the following is true?
(a)powlaer
(b)pow
(c)power
(d)Compiler error
(e)None of these
(13)

```
void main(){

        int a=270;
        char *p;
        p=(char *)&a;
        clrscr();
        printf("%d",*p);
        getch();
}
```
Which of the following is true?
(a)270
(b)address of variable a
(c)16
(d)Compiler error
(e)None of these
(14)
What is missing statement of in the following program?
```
void main(){
        int sort(int,int);
        int I;
        i=sort(5,6);
}
int sort(int a,int b){
        int c;
        c=a;
        a=b;
        b=c;
        return a;
}
```
(15)Write following in term of if and else:
```
void main(){
        int a=1,b=2,c=3;
        clrscr();
        if(a==5&&b==6&&c==7)
        printf("india");
        else
        printf("pak");
        getch();
```

```
}
```
(16)
Draw memory representation of
```
struct xxx{
        char a;
        int b;
        char c;
};
```
(17)
Write the following program in term of **switch** and **case**?
```
void main()
{
        int a=3;
        if(x>2){
        printf("INDIA IS BEST");
        }
        else{
        printf("PAK IS BEST");
        }
}
```
(18)
```
void main(){
        int far *a=(int far*)0x50000011;
        int far *b=(int far*)0x50010001;
        int huge *c=(int huge*)0x50000011;
        int huge *d=(int huge*)0x50010001;
        clrscr();
        if(a==b)
        printf("I know C");
        else
        printf("I don't know C");
        if(c==d)
        printf("\nI know C");
        else
        printf("\nI don't know C");
        getch();
}
```
Which of the following is true?
(a)I know C
I Know C
(b)I know C
I don't know C
(c)I don't know C
I know C
(d)Compiler error
(e)None of these
(19)
```
#define power(a) #a
void main(){
        clrscr();
```

```
        printf("%d",*power(432));
        getch();
}
```
Which of the following is true?
(a)*"432"
(b)432
(c)16
(d)32
(e)Compiler error
(20)
```
void main(){
        int arr[]={1,2,3,4,5,6};
        void xxx(int[5]);
        xxx(arr);
        getch();
}
void xxx(int ch[5]){
        clrscr();
        printf("%d",-1[ch]);
}
```
Which of the following true?
(a)2
(b)-2
(c)3
(d)-3
(e)Compiler error
(21)
What is difference between a, b, c and in following declaration ?
```
#define xxx char *
typedef char * yyy;
void main(){
        yyy a,b;
        xxx c,d;
}
```
(22)
Write a c program to find the HCF of any two numbers?
(23)
```
void main(){
        int a=5;{
        a++;
        }
        clrscr();
        printf("%d",a);
        getch();
}
```
Which of the following is true?
(a)5
(b)6
(c)7
(d)Compiler error

(e)None of these
(24)

```
void main(){
        int a=5;{
        int a=7;
        a++;
        printf(“%d”,a);
        }
        clrscr();
        printf("%d",a);
        getch();
}
```

Which of the following is true?
(a)5 7
(b)5 8
(c)8 5
(d)7 5
(e)Compiler error

Answer:
(1)(b)
(2)(c)
(3)(c)
(4) See in explanation.
(5) See in explanation.
(6) See in explanation.
(7) See in explanation.
(8)(c)
(9)(c)
(10)(b)
(11) See in explanation.
(12) See in explanation.
(13)(b)
(14)(c)
Explanation:
(11) 3.8f is **float** constant, 3.8 is **double** constant and 3.8L is **long double** constant .Here are finding size of **double** constantan which is 8.
(12) \b escape sequence back the cursor one position left .We are using two /b so after writing str1 cursor is at the position of l of powal .So when it write er it will override the la so output will be power.
(14) Function sort returning a value but we are not using **return** value so there is wastage of two byte memory. So missing statement is, there should statement which uses the **return** value.
(15)

```
void main(){
        int a=1,b=2,c=3;
        clrscr();
        if(a==1){
        if(b==2){
```

```
        if(c==3){
        printf("india");
        }
        else{
        printf("pak");
        }
        }
        else{
        printf("pak");
        }
        }
        else{
        printf("pak");
        }
        getch();
}
```

(17)**if** condition always **return** two value.
1 **if** condition is true.
0 **if** condition is false.
So program is

```
void main(){
        int x=3;
        switch(x>2){
        case 0:printf("India is best");
        break;
        case 1:printf("Pak is best");
        }
        getch();
}
```

(18)
far pointer always compare its whole far address. Since both or not equal so first output is: I don't know C
Huge pointer always compare its physical address both c and d are representing same physical address so a and b are equal.
(19) # is string zinging operator. It makes the string constant of any data. So 432 is converted into "432" by macro power .Now *"432" means first **char** which is 4.Since we are using %d so it will print ASCII value of **char** 4 i.e. 52
(20) We are passing the array by xxx function. 1[ch] means *(ch+1) which is ch[1] =2.
(21) Both and b are **char** * type but c is **char** * type **while** d is **char** type.
(22)

```
void main(){
        int a,b,c;
        scanf("%d%d%d",a,b,c);
        clrscr();
        while((c=a%b)!=0){
        a=b;
        b=c;
        }
        printf("%d",b);
        getch();
}
```

(24) Scope of the **auto** variable is within { } **if** it is declared in { }.Also local variable has more priority than global variable.

25. Which of the following statements should be used to obtain a remainder after dividing 3.14 by 2.1 ?

A.rem = 3.14 % 2.1;
B.rem = modf(3.14, 2.1);
C.rem = fmod(3.14, 2.1);
D.Remainder cannot be obtain in floating point division.

Answer: Option **C**

Explanation:

fmod(x,y) - Calculates x modulo y, the remainder of x/y.
This function is the same as the modulus operator. But *fmod()* performs floating point divisions.

26.

What are the types of linkages?

A.Internal and External
B.External, Internal and None
C.External and None
D.Internal
Answer & Explanation

Answer: Option **B**

Explanation:

External Linkage-> means global, non-static variables and functions.
Internal Linkage-> means static variables and functions with file scope.
None Linkage-> means Local variables.

27.

Which of the following special symbol allowed in a variable name?

A.* (asterisk)
B.| (pipeline)
C.- (hyphen)
D._ (underscore)
Answer & Explanation

Answer: Option **D**

Explanation:

Variable names in C are made up of letters (upper and lower case) and digits. The underscore character ("_") is also permitted. Names must not begin with a digit.

Examples of valid (but not very descriptive) C variable names:
=> foo
=> Bar
=> BAZ
=> foo_bar
=> _foo42
=> _
=> QuUx

28.

Is there any difference between following declarations?

1 : extern int fun();
2 : int fun();
A.Both are identical
B.No difference, except *extern int fun();* is probably in another file
C.*int fun();* is overrided with *extern int fun();*
D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

extern int fun(); declaration in C is to indicate the existence of a global function and it is defined externally to the current module or in another file.

int fun(); declaration in C is to indicate the existence of a function inside the current module or in the same file.

29.#include<stdio.h>
```
int main()
{
   enum status { pass, fail, atkt};
   enum status stud1, stud2, stud3;
   stud1 = pass;
   stud2 = atkt;
   stud3 = fail;
   printf("%d, %d, %d\n", stud1, stud2, stud3);
   return 0;
}
```
A.0, 1, 2 B.1, 2, 3
C.0, 2, 1 D.1, 3, 2
Answer & Explanation

Answer: Option **C**

Explanation:

enum takes the format like {0,1,2..) so *pass*=0, *fail*=1, *atkt*=2

stud1 = *pass* (value is 0)

stud2 = *atkt* (value is 2)

stud3 = *fail* (value is 1)

Hence it prints 0, 2, 1

30.

What will be the output of the program in 16 bit platform (Turbo C under DOS)?

```
#include<stdio.h>
int main()
{
   extern int i;
   i = 20;
   printf("%d\n", sizeof(i));
   return 0;
}
```

A.2
B.4
C.vary from compiler
D.Linker Error : Undefined symbol *'i'*
Answer & Explanation

Answer: Option **D**

Explanation:

Linker Error : Undefined symbol *'i'*
The statement *extern int i* specifies to the compiler that the memory for *'i'* is allocated in some other program and that address will be given to the current program at the time of linking. But linker finds that no other variable of name *'i'* is available in any other program with memory space allocated for it. Hence a linker error has occurred.

31.

What is the output of the program?

```
#include<stdio.h>
int main()
{
   extern int a;
   printf("%d\n", a);
   return 0;
}
int a=20;
```

A.20 B.0
C.Garbage Value D.Error
Answer & Explanation

Answer: Option **A**

Explanation:

extern int a; indicates that the variable *a* is defined elsewhere, usually in a separate source code module.

printf("%d\n", a); it prints the value of local variable *int a = 20*. Because, whenever there is a conflict between local variable and global variable, local variable gets the highest priority. So it prints 20.

32.

What is the output of the program in Turbo C (in DOS 16-bit OS)?

```
#include<stdio.h>
int main()
{
   char *s1;
   char far *s2;
   char huge *s3;
   printf("%d, %d, %d\n", sizeof(s1), sizeof(s2), sizeof(s3));
   return 0;
}
```

A.2, 4, 6 B.4, 4, 2
C.2, 4, 4 D.2, 2, 2
Answer & Explanation

Answer: Option **C**

Explanation:

Any pointer size is 2 bytes. (only 16-bit offset)
So, *char *s1* = 2 bytes.
So, *char far *s2;* = 4 bytes.
So, *char huge *s3;* = 4 bytes.
A far, huge pointer has two parts: a 16-bit segment value and a 16-bit offset value.

Since C is a compiler dependent language, it may give different output in other platforms. The above program works fine in Windows (TurboC), but error in Linux (GCC Compiler).

33.

What is the output of the program

```
#include<stdio.h>
```

```
int main()
{
   struct emp
   {
      char name[20];
      int age;
      float sal;
   };
   struct emp e = {"Tiger"};
   printf("%d, %f\n", e.age, e.sal);
   return 0;
}
```

A.0, 0.000000
B.Garbage values
C.Error
D.None of above

Answer & Explanation

Answer: Option **A**

Explanation:

When an automatic structure is partially initialized remaining elements are initialized to 0(zero).

34.

What will be the output of the program?

```
#include<stdio.h>
int X=40;
int main()
{
   int X=20;
   printf("%d\n", X);
   return 0;
}
```

A.20
B.40
C.Error
D.No Output

Answer & Explanation

Answer: Option **A**

Explanation:

Whenever there is conflict between a local variable and global variable, the local variable gets priority.

35.

What is the output of the program

```
#include<stdio.h>
int main()
{
   int x = 10, y = 20, z = 5, i;
   i = x < y < z;
   printf("%d\n", i);
   return 0;
}
```

A.0 B.1
C.Error D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

Since *x < y* turns to be TRUE it is replaced by 1. Then *1 < z* is compared and to be *TRUE*. The 1 is assigned to *i*.

36.

What is the output of the program

```
#include<stdio.h>
int main()
{
   extern int fun(float);
   int a;
   a = fun(3.14);
   printf("%d\n", a);
   return 0;
}
int fun(int aa)
{
   return (int)++aa;
}
```

A.3 B.3.14
C.0 D.4
E. Compile Error
Answer & Explanation

Answer: Option **E**

Explanation:

2 Errors
1. Type mismatch in redeclaration of *fun*
2. Type mismatch in parameter *aa*

37.

What is the output of the program

```
#include<stdio.h>
int main()
{
   int a[5] = {2, 3};
   printf("%d, %d, %d\n", a[2], a[3], a[4]);
   return 0;
}
```

A.Garbage Values
B.2, 3, 3
C.3, 2, 2
D.0, 0, 0

Answer & Explanation

Answer: Option **D**

Explanation:

When an automatic array is partially initialized, the remaining elements are initialized to 0.

38.

What is the output of the program?

```
#include<stdio.h>
int main()
{
   union a
   {
      int i;
      char ch[2];
   };
   union a u;
   u.ch[0] = 3;
   u.ch[1] = 2;
   printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i);
   return 0;
}
```

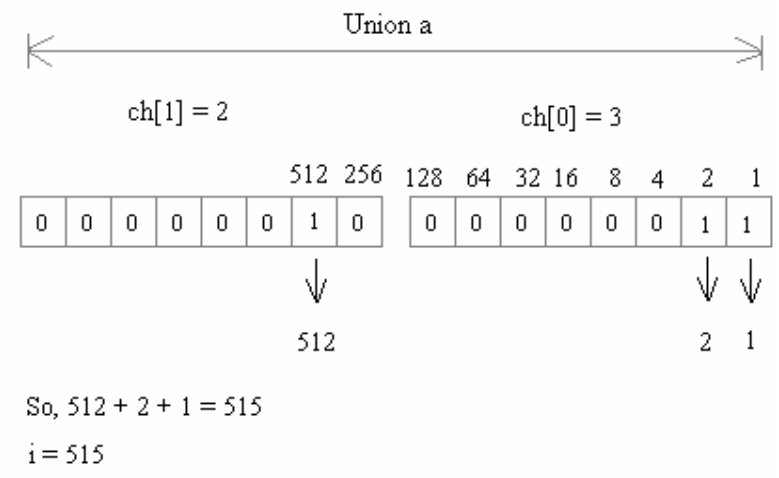
A.3, 2, 515
B.515, 2, 3
C.3, 2, 5
D.None of these

Answer & Explanation

Answer: Option **A**

Explanation:

printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i); It prints the value of *u.ch[0] = 3, u.ch[1] = 2* and it prints the value of *u.i* means the value of entire union size.



So the output is 3, 2, 515.

39.

In the following program how long will the *for* loop get executed?

```
#include<stdio.h>
int main()
{
   int i=5;
   for(;scanf("%s", &i); printf("%d\n", i));
   return 0;
}
```

A.The *for* loop would not get executed at all
B.The *for* loop would get executed only once
C.The *for* loop would get executed 5 times
D.The *for* loop would get executed infinite times
Answer & Explanation

Answer: Option **D**

Explanation:

During the *for* loop execution *scanf()* ask input and then *printf()* prints that given input. This process will be continued repeatedly because, *scanf()* returns the number of input given, the condition is always true(user gives a input means it reurns '1').

Hence this *for* loop would get executed infinite times.

40.

What will be the output of the program?

```
#include<stdio.h>
int main()
```

```
{
   int X=40;
   {
      int X=20;
      printf("%d ", X);
   }
   printf("%d\n", X);
   return 0;
}
```

A.40 40
B.20 40
C.20
D.Error

Answer & Explanation

Answer: Option **B**

Explanation:

In case of a conflict between a local variable and global variable, the local variable gets priority.

41.Point out the error in the following program (if it is compiled with Turbo C compiler).

```
#include<stdio.h>
int main()
{
   display();
   return 0;
}
void display()
{
   printf("IndiaBIX.com");
}
```

A.No error
B.*display()* doesn't get invoked
C.*display()* is called before it is defined
D.None of these

Answer & Explanation

Answer: Option **C**

Explanation:

In this program the compiler will not know that the function *display()* exists. So, the compiler will generate "Type mismatch in redeclaration of function *display()*".

To over come this error, we have to add function prototype of function *display()*.
Another way to overcome this error is to define the function *display()* before the *int main();* function.

```
#include<stdio.h>
void display(); /* function prototype */

int main()
{
   display();
   return 0;
}
void display()
{
   printf("IndiaBIX.com");
}
```

Output: IndiaBIX.com

Note: This problem will not occur in modern compilers (this problem occurs in TurboC but not in GCC).

42.Point out the error in the following program.

```
#include<stdio.h>
int main()
{
   void v = 0;

   printf("%d", v);

   return 0;
}
```

A.Error: Declaration syntax error '*v*' (or) Size of *v* is unknown or zero.
B.Program terminates abnormally.
C.No error.
D.None of these.
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

43.Point out the error in the following program.

```
#include<stdio.h>
struct emp
{
   char name[20];
   int age;
};
int main()
```

```
{
   emp int xx;
   int a;
   printf("%d\n", &a);
   return 0;
}
```

A.Error: in *printf*
B.Error: in *emp int xx;*
C.No error.
D.None of these.

Answer & Explanation

Answer: Option **B**

Explanation:

There is an error in the line *emp int xx;*

To overcome this error, remove the *int* and add the *struct* at the begining of *emp int xx;*

```
#include<stdio.h>
struct emp
{
   char name[20];
   int age;
};
int main()
{
   struct emp xx;
   int a;
   printf("%d\n", &a);
   return 0;
}
```

44.Which of the following is correct about *err* used in the declaration given below?

```
typedef enum error { warning, test, exception } err;
```

A.It is a *typedef* for *enum error*.
B.It is a variable of type *enum error*.
C.The statement is erroneous.
D.It is a structure.
Answer & Explanation

Answer: Option **A**

Explanation:

A *typedef* gives a new name to an existing data type.
So *err* is a new name for *enum error*.

45.Point out the error in the following program.

```
#include<stdio.h>
int main()
{
   int (*p)() = fun;
   (*p)();
   return 0;
}
int fun()
{
   printf("IndiaBix.com\n");
   return 0;
}
```

A.Error: in *int(*p)() = fun;*
B.Error: *fun()* prototype not defined
C.No error
D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

The compiler will not know that the function *int fun()* exists. So we have to define the function prototype of *int fun();*
To overcome this error, see the below program

```
#include<stdio.h>
int fun(); /* function prototype */

int main()
{
   int (*p)() = fun;
   (*p)();
   return 0;
}
int fun()
{
   printf("IndiaBix.com\n");
   return 0;
}
```

46.What is (void*)0?

A.Representation of NULL pointer
B.Representation of void pointer
C.Error
D.None of above
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

47.

Can you combine the following two statements into one?

char *p;
p = (char*) malloc(100);
A.char p = *malloc(100);
B.char *p = (char) malloc(100);
C.char *p = (char*)malloc(100);
D.char *p = (char *)(malloc*)(100);
Answer & Explanation

Answer: Option **C**

Explanation:

No answer description available for this question.

48.

In which header file is the NULL macro defined?

A.stdio.h
B.stddef.h
C.stdio.h and stddef.h
D.math.h
Answer & Explanation

Answer: Option **C**

Explanation:

The macro "NULL" is defined in locale.h, stddef.h, stdio.h, stdlib.h, string.h, time.h, and wchar.h.

49.

How many bytes are occupied by *near, far* and *huge* pointers (DOS)?

A.near=2 far=4 huge=4
B.near=4 far=8 huge=8
C.near=2 far=4 huge=8
D.near=4 far=4 huge=8
Answer & Explanation

Answer: Option **A**

Explanation:

near=2, *far=4* and *huge=4* pointers exist only under DOS. Under windows and Linux every pointers is 4 bytes long.

50.

If a variable is a pointer to a structure, then which of the following operator is used to access data members of the structure through the pointer variable?

A.. B.&
C.* D.->
Answer & Explanation

Answer: Option **D**

Explanation:

No answer description available for this question.

51.

What would be the equivalent pointer expression for referring the array element *a[i][j][k][l]*

A.((((a+i)+j)+k)+l) B.*(*(*(*(a+i)+j)+k)+l)
C.(((a+i)+j)+k+l) D.((a+i)+j+k+l)
Answer & Explanation

Answer: Option **B**

Explanation:

No answer description available for this question.

52.

A pointer is

A.A keyword used to create variables
B.A variable that stores address of an instruction
C.A variable that stores address of other variable
D.All of the above
Answer & Explanation

Answer: Option **C**

Explanation:

No answer description available for this question.

53.

The operator used to get value at address stored in a pointer variable is

A.* B.&
C.&& D.||
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

54.

What is (void*)0?

A.Representation of NULL pointer
B.Representation of void pointer
C.Error
D.None of above
Answer & Explanation

Answer: Option **A**

55.Can you combine the following two statements into one?

char *p;
p = (char*) malloc(100);
A.char p = *malloc(100);
B.char *p = (char) malloc(100);
C.char *p = (char*)malloc(100);
D.char *p = (char *)(malloc*)(100);
Answer & Explanation

Answer: Option **C**

56) Character set of C language contains ?

5. Alphabets
6. Digits
7. Special Symbols
8. All of these

56) A Variable name can have ?

5. Any special symbol
6. blank spaces
7. double

8. char

57) In C language , one of the following is not a valid data type :

5. long
6. float
7. double
8. char

58) The format string %lf is used for ?

5. float
6. double
7. unsigned int
8. long double

59) A variable of type unsigned int can have a value in the range ?

5. -32768 to +32767
6. 0 to 32767
7. 0 to 65535
8. -32767 to +32767

60) Which data type is not a primary data type ?

5. int
6. array
7. float
8. char

61) Which of the format string is not valid ?

5. %ld
6. %lf
7. %lu
8. %lc

62) Which is the valid string data ?

5. 'A'
6. A
7. "A"
8. none of these

63) How much memory is required to store a value of type double ?

5. 4 bytes
6. 6 bytes
7. 8 bytes
8. 10 bytes

64) The modifier which is used to declare a variable as constant ?

6. short
7. signed
8. unsigned
9. const
10. **Answers :**

1)	All of these	6)	array
2)	underscore	7)	%lc
3)	long	8)	"A"
4)	double	9)	8 bytes
5)	0 to 65535	10)	const

(67)

```
void main(){
        clrscr();
        printf("%d",sizeof(3.8));
        getch();
}
```

Which of the following is true?
(a)4
(b)8
(c)10
(d)Compiler error
(e)None of these

(68)

```
void main(){
        char *str1="powla";
        char *str2="er";
        clrscr();
        printf("%s\b\b%s",str1,str2);
        getch();
}
```

Which of the following is true?
(a)powlaer
(b)pow
(c)power
(d)Compiler error
(e)None of these
(69)

```
void main(){

        int a=270;
        char *p;
        p=(char *)&a;
```

```
        clrscr();
        printf("%d",*p);
        getch();
}
```
Which of the following is true?
(a)270
(b)address of variable a
(c)16
(d)Compiler error
(e)None of these
(70)
What is missing statement of in the following program?
```
void main(){
        int sort(int,int);
        int I;
        i=sort(5,6);
}
int sort(int a,int b){
        int c;
        c=a;
        a=b;
        b=c;
        return a;
}
```
(71)Write following in term of if and else:
```
void main(){
        int a=1,b=2,c=3;
        clrscr();
        if(a==5&&b==6&&c==7)
        printf("india");
        else
        printf("pak");
        getch();
}
```
(72)
Draw memory representation of
```
struct xxx{
        char a;
        int b;
        char c;
};
```
(73)
Write the following program in term of **switch** and **case**?
```
void main()
{
        int a=3;
        if(x>2){
        printf("INDIA IS BEST");
        }
        else{
```

```
        printf("PAK IS BEST");
        }
}
```
(74)
```
void main(){
        int far *a=(int far*)0x50000011;
        int far *b=(int far*)0x50010001;
        int huge *c=(int huge*)0x50000011;
        int huge *d=(int huge*)0x50010001;
        clrscr();
        if(a==b)
        printf("I know C");
        else
        printf("I don't know C");
        if(c==d)
        printf("\nI know C");
        else
        printf("\nI don't know C");
        getch();
}
```
Which of the following is true?
(a)I know C
I Know C
(b)I know C
I don't know C
(c)I don't know C
I know C
(d)Compiler error
(e)None of these
(75)
```
#define power(a) #a
void main(){
        clrscr();
        printf("%d",*power(432));
        getch();
}
```
Which of the following is true?
(a)*"432"
(b)432
(c)16
(d)32
(e)Compiler error
(20)
```
void main(){
        int arr[]={1,2,3,4,5,6};
        void xxx(int[5]);
        xxx(arr);
        getch();
}
void xxx(int ch[5]){
```

```
        clrscr();
        printf("%d",-1[ch]);
}
```
Which of the following true?
(a)2
(b)-2
(c)3
(d)-3
(e)Compiler error
(121)
What is difference between a, b, c and in following declaration ?
```
#define xxx char *
typedef char * yyy;
void main(){
        yyy a,b;
        xxx c,d;
}
```
(122)
Write a c program to find the HCF of any two numbers?
(123)
```
void main(){
        int a=5;{
        a++;
        }
        clrscr();
        printf("%d",a);
        getch();
}
```
Which of the following is true?
(a)5
(b)6
(c)7
(d)Compiler error
(e)None of these
(124)
```
void main(){
        int a=5;{
        int a=7;
        a++;
        printf("%d",a);
        }
        clrscr();
        printf("%d",a);
        getch();
}
```
Which of the following is true?
(a)5 7
(b)5 8
(c)8 5
(d)7 5

(e)Compiler error

Answer:
(1)(b)
(2)(c)
(3)(c)
(4) See in explanation.
(5) See in explanation.
(6) See in explanation.
(7) See in explanation.
(8)(c)
(9)(c)
(10)(b)
(11) See in explanation.
(12) See in explanation.
(13)(b)
(14)(c)
Explanation:
(11) 3.8f is **float** constant, 3.8 is **double** constant and 3.8L is **long double** constant .Here are finding size of **double** constantan which is 8.
(12) \b escape sequence back the cursor one position left .We are using two /b so after writing str1 cursor is at the position of l of powal .So when it write er it will override the la so output will be power.
(14) Function sort returning a value but we are not using **return** value so there is wastage of two byte memory. So missing statement is, there should statement which uses
the **return** value.
(15)

```
void main(){
        int a=1,b=2,c=3;
        clrscr();
        if(a==1){
        if(b==2){
        if(c==3){
        printf("india");
        }
        else{
        printf("pak");
        }
        }
        else{
        printf("pak");
        }
        }
        else{
        printf("pak");
        }
        getch();
}
```

(17)**if** condition always **return** two value.

1 **if** condition is true.
0 **if** condition is false.
So program is

```
void main(){
        int x=3;
        switch(x>2){
        case 0:printf("India is best");
        break;
        case 1:printf("Pak is best");
        }
        getch();
}
```

(18)
far pointer always compare its whole far address. Since both or not equal so first output is: I don't know C
Huge pointer always compare its physical address both c and d are representing same physical address so a and b are equal.
(19) # is string zinging operator. It makes the string constant of any data. So 432 is converted into "432" by macro power .Now *"432" means first **char** which is 4.Since we are using %d so it will print ASCII value of **char** 4 i.e. 52
(20) We are passing the array by xxx function. 1[ch] means *(ch+1) which is ch[1] =2.
(21) Both and b are **char** * type but c is **char** * type **while** d is **char** type.
(22)

```
void main(){
        int a,b,c;
        scanf("%d%d%d",a,b,c);
        clrscr();
        while((c=a%b)!=0){
        a=b;
        b=c;
        }
        printf("%d",b);
        getch();
}
```

(24) Scope of the **auto** variable is within {} **if** it is declared in {}.Also local variable has more priority than global variable.

25. Which of the following statements should be used to obtain a remainder after dividing 3.14 by 2.1 ?

A.rem = 3.14 % 2.1;
B.rem = modf(3.14, 2.1);
C.rem = fmod(3.14, 2.1);
D.Remainder cannot be obtain in floating point division.

Answer: Option **C**

Explanation:

fmod(x,y) - Calculates x modulo y, the remainder of x/y.
This function is the same as the modulus operator. But *fmod()* performs floating point divisions.

26.

What are the types of linkages?

A.Internal and External
B.External, Internal and None
C.External and None
D.Internal
Answer & Explanation

Answer: Option **B**

Explanation:

External Linkage-> means global, non-static variables and functions.
Internal Linkage-> means static variables and functions with file scope.
None Linkage-> means Local variables.

27.

Which of the following special symbol allowed in a variable name?

A.* (asterisk)
B.| (pipeline)
C.- (hyphen)
D._ (underscore)
Answer & Explanation

Answer: Option **D**

Explanation:

Variable names in C are made up of letters (upper and lower case) and digits. The underscore character ("_") is also permitted. Names must not begin with a digit.

Examples of valid (but not very descriptive) C variable names:
=> foo
=> Bar
=> BAZ
=> foo_bar
=> _foo42
=> _
=> QuUx

28.

Is there any difference between following declarations?

```
1 : extern int fun();
```

2 : int fun();

A.Both are identical
B.No difference, except *extern int fun();* is probably in another file
C.*int fun();* is overrided with *extern int fun();*
D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

extern int fun(); declaration in C is to indicate the existence of a global function and it is defined externally to the current module or in another file.

int fun(); declaration in C is to indicate the existence of a function inside the current module or in the same file.

29.#include<stdio.h>

```
int main()
{
   enum status { pass, fail, atkt};
   enum status stud1, stud2, stud3;
   stud1 = pass;
   stud2 = atkt;
   stud3 = fail;
   printf("%d, %d, %d\n", stud1, stud2, stud3);
   return 0;
}
```

A.0, 1, 2 B.1, 2, 3
C.0, 2, 1 D.1, 3, 2
Answer & Explanation

Answer: Option **C**

Explanation:

enum takes the format like {0,1,2..) so *pass*=0, *fail*=1, *atkt*=2

stud1 = *pass* (value is 0)

stud2 = *atkt* (value is 2)

stud3 = *fail* (value is 1)

Hence it prints 0, 2, 1

30.

What will be the output of the program in 16 bit platform (Turbo C under DOS)?

```
#include<stdio.h>
int main()
{
   extern int i;
   i = 20;
   printf("%d\n", sizeof(i));
   return 0;
}
```

A.2
B.4
C.vary from compiler
D.Linker Error : Undefined symbol *'i'*
Answer & Explanation

Answer: Option **D**

Explanation:

Linker Error : Undefined symbol *'i'*
The statement *extern int i* specifies to the compiler that the memory for *'i'* is allocated in some other program and that address will be given to the current program at the time of linking. But linker finds that no other variable of name *'i'* is available in any other program with memory space allocated for it. Hence a linker error has occurred.

31.

What is the output of the program?

```
#include<stdio.h>
int main()
{
   extern int a;
   printf("%d\n", a);
   return 0;
}
int a=20;
```

A.20 B.0
C.Garbage Value D.Error
Answer & Explanation

Answer: Option **A**

Explanation:

extern int a; indicates that the variable *a* is defined elsewhere, usually in a separate source code module.

printf("%d\n", a); it prints the value of local variable *int a = 20*. Because, whenever there is a conflict between local variable and global variable, local variable gets the highest priority. So it prints 20.

32.

What is the output of the program in Turbo C (in DOS 16-bit OS)?

```
#include<stdio.h>
int main()
{
   char *s1;
   char far *s2;
   char huge *s3;
   printf("%d, %d, %d\n", sizeof(s1), sizeof(s2), sizeof(s3));
   return 0;
}
```

A.2, 4, 6
B.4, 4, 2
C.2, 4, 4
D.2, 2, 2

Answer & Explanation

Answer: Option **C**

Explanation:

Any pointer size is 2 bytes. (only 16-bit offset)
So, *char *s1* = 2 bytes.
So, *char far *s2;* = 4 bytes.
So, *char huge *s3;* = 4 bytes.
A far, huge pointer has two parts: a 16-bit segment value and a 16-bit offset value.

Since C is a compiler dependent language, it may give different output in other platforms.
The above program works fine in Windows (TurboC), but error in Linux (GCC Compiler).

33.

What is the output of the program

```
#include<stdio.h>
int main()
{
   struct emp
   {
      char name[20];
      int age;
      float sal;
   };
   struct emp e = {"Tiger"};
   printf("%d, %f\n", e.age, e.sal);
   return 0;
}
```

A.0, 0.000000
B.Garbage values
C.Error
D.None of above

Answer & Explanation

Answer: Option **A**

Explanation:

When an automatic structure is partially initialized remaining elements are initialized to 0(zero).

34.

What will be the output of the program?

```
#include<stdio.h>
int X=40;
int main()
{
   int X=20;
   printf("%d\n", X);
   return 0;
}
```

A.20 B.40
C.Error D.No Output
Answer & Explanation

Answer: Option **A**

Explanation:

Whenever there is conflict between a local variable and global variable, the local variable gets priority.

35.

What is the output of the program

```
#include<stdio.h>
int main()
{
   int x = 10, y = 20, z = 5, i;
   i = x < y < z;
   printf("%d\n", i);
   return 0;
}
```

A.0 B.1
C.Error D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

Since *x* < *y* turns to be TRUE it is replaced by 1. Then *1* < *z* is compared and to be *TRUE*. The 1 is assigned to *i*.

36.

What is the output of the program

```
#include<stdio.h>
int main()
{
   extern int fun(float);
   int a;
   a = fun(3.14);
   printf("%d\n", a);
   return 0;
}
int fun(int aa)
{
   return (int)++aa;
}
```

A.3 B.3.14
C.0 D.4
E. Compile Error
Answer & Explanation

Answer: Option **E**

Explanation:

2 Errors
1. Type mismatch in redeclaration of *fun*
2. Type mismatch in parameter *aa*

37.

What is the output of the program

```
#include<stdio.h>
int main()
{
   int a[5] = {2, 3};
   printf("%d, %d, %d\n", a[2], a[3], a[4]);
   return 0;
}
```

A.Garbage Values B.2, 3, 3
C.3, 2, 2 D.0, 0, 0
Answer & Explanation

Answer: Option **D**

Explanation:

When an automatic array is partially initialized, the remaining elements are initialized to 0.

38.

What is the output of the program?

```
#include<stdio.h>
int main()
{
   union a
   {
      int i;
      char ch[2];
   };
   union a u;
   u.ch[0] = 3;
   u.ch[1] = 2;
   printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i);
   return 0;
}
```

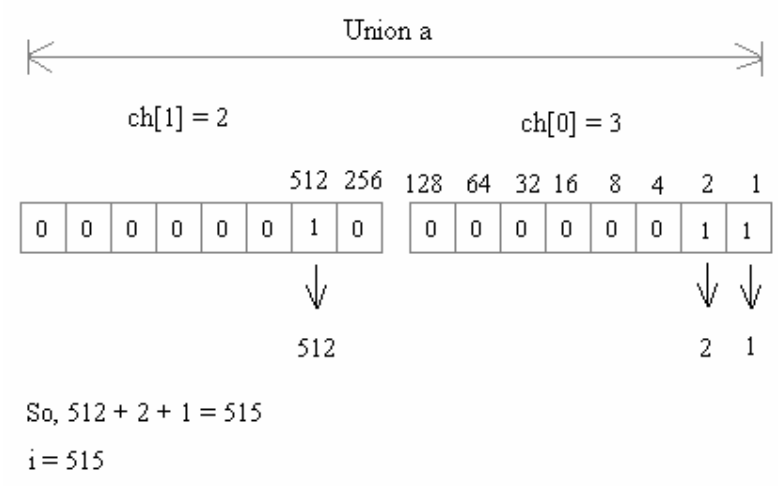
A.3, 2, 515
B.515, 2, 3
C.3, 2, 5
D.None of these

Answer & Explanation

Answer: Option **A**

Explanation:

printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i); It prints the value of *u.ch[0] = 3, u.ch[1] = 2* and it prints the value of *u.i* means the value of entire union size.



So the output is 3, 2, 515.

39.

In the following program how long will the *for* loop get executed?

```
#include<stdio.h>
int main()
{
   int i=5;
   for(;scanf("%s", &i); printf("%d\n", i));
   return 0;
}
```

A.The *for* loop would not get executed at all
B.The *for* loop would get executed only once
C.The *for* loop would get executed 5 times
D.The *for* loop would get executed infinite times
Answer & Explanation

Answer: Option **D**

Explanation:

During the *for* loop execution *scanf()* ask input and then *printf()* prints that given input. This process will be continued repeatedly because, *scanf()* returns the number of input given, the condition is always true(user gives a input means it reurns '1').

Hence this *for* loop would get executed infinite times.

40.

What will be the output of the program?

```
#include<stdio.h>
int main()
{
   int X=40;
   {
      int X=20;
      printf("%d ", X);
   }
   printf("%d\n", X);
   return 0;
}
```

A.40 40
B.20 40
C.20
D.Error
Answer & Explanation

Answer: Option **B**

Explanation:

In case of a conflict between a local variable and global variable, the local variable gets priority.

41.Point out the error in the following program (if it is compiled with Turbo C compiler).

```
#include<stdio.h>
int main()
{
   display();
   return 0;
}
void display()
{
   printf("IndiaBIX.com");
}
```

A.No error
B.*display()* doesn't get invoked
C.*display()* is called before it is defined
D.None of these
Answer & Explanation

Answer: Option **C**

Explanation:

In this program the compiler will not know that the function *display()* exists. So, the compiler will generate "Type mismatch in redeclaration of function *display()*".

To over come this error, we have to add function prototype of function *display()*.
Another way to overcome this error is to define the function *display()* before the *int main();* function.

```
#include<stdio.h>
void display(); /* function prototype */

int main()
{
   display();
   return 0;
}
void display()
{
   printf("IndiaBIX.com");
}
```

Output: IndiaBIX.com

Note: This problem will not occur in modern compilers (this problem occurs in TurboC but not in GCC).

42.Point out the error in the following program.

```
#include<stdio.h>
int main()
{
   void v = 0;

   printf("%d", v);

   return 0;
}
```

A.Error: Declaration syntax error '*v*' (or) Size of *v* is unknown or zero.
B.Program terminates abnormally.
C.No error.
D.None of these.
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

43.Point out the error in the following program.

```
#include<stdio.h>
struct emp
{
   char name[20];
   int age;
};
int main()
{
   emp int xx;
   int a;
   printf("%d\n", &a);
   return 0;
}
```

A.Error: in *printf*
B.Error: in *emp int xx;*
C.No error.
D.None of these.
Answer & Explanation

Answer: Option **B**

Explanation:

There is an error in the line *emp int xx;*

To overcome this error, remove the *int* and add the *struct* at the begining of *emp int xx;*

```
#include<stdio.h>
struct emp
{
   char name[20];
   int age;
};
int main()
{
   struct emp xx;
   int a;
   printf("%d\n", &a);
   return 0;
}
```

44.Which of the following is correct about *err* used in the declaration given below?

```
typedef enum error { warning, test, exception } err;
```

A.It is a *typedef* for *enum error*.
B.It is a variable of type *enum error*.
C.The statement is erroneous.
D.It is a structure.
Answer & Explanation

Answer: Option **A**

Explanation:

A *typedef* gives a new name to an existing data type.
So *err* is a new name for *enum error*.

45.Point out the error in the following program.

```
#include<stdio.h>
int main()
{
   int (*p)() = fun;
   (*p)();
   return 0;
}
int fun()
{
   printf("IndiaBix.com\n");
   return 0;
}
```

A.Error: in *int(*p)() = fun;*
B.Error: *fun()* prototype not defined
C.No error
D.None of these
Answer & Explanation

Answer: Option **B**

Explanation:

The compiler will not know that the function *int fun()* exists. So we have to define the function prototype of *int fun();*
To overcome this error, see the below program

```
#include<stdio.h>
int fun(); /* function prototype */

int main()
{
   int (*p)() = fun;
   (*p)();
   return 0;
}
int fun()
{
   printf("IndiaBix.com\n");
   return 0;
}
```

46.What is (void*)0?

A.Representation of NULL pointer
B.Representation of void pointer
C.Error
D.None of above
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

47.

Can you combine the following two statements into one?

```
char *p;
p = (char*) malloc(100);
```

A.char p = *malloc(100);
B.char *p = (char) malloc(100);
C.char *p = (char*)malloc(100);
D.char *p = (char *)(malloc*)(100);
Answer & Explanation

Answer: Option **C**

Explanation:

No answer description available for this question.

48.

In which header file is the NULL macro defined?

A.stdio.h
B.stddef.h
C.stdio.h and stddef.h
D.math.h
Answer & Explanation

Answer: Option **C**

Explanation:

The macro "NULL" is defined in locale.h, stddef.h, stdio.h, stdlib.h, string.h, time.h, and wchar.h.

49.

How many bytes are occupied by *near, far* and *huge* pointers (DOS)?

A.near=2 far=4 huge=4
B.near=4 far=8 huge=8
C.near=2 far=4 huge=8
D.near=4 far=4 huge=8
Answer & Explanation

Answer: Option **A**

Explanation:

near=2, *far=4* and *huge=4* pointers exist only under DOS. Under windows and Linux every pointers is 4 bytes long.

50.

If a variable is a pointer to a structure, then which of the following operator is used to access data members of the structure through the pointer variable?

A..
B.&
C.*
D.->
Answer & Explanation

Answer: Option **D**

Explanation:

No answer description available for this question.

51.

What would be the equivalent pointer expression for referring the array element *a[i][j][k][l]*

A.((((a+i)+j)+k)+l)
B.*(*(*(*(a+i)+j)+k)+l)
C.(((a+i)+j)+k+l)
D.((a+i)+j+k+l)
Answer & Explanation

Answer: Option **B**

Explanation:

No answer description available for this question.

52.

A pointer is

A.A keyword used to create variables
B.A variable that stores address of an instruction
C.A variable that stores address of other variable
D.All of the above
Answer & Explanation

Answer: Option **C**

Explanation:

No answer description available for this question.

53.

The operator used to get value at address stored in a pointer variable is

A.*
B.&
C.&&
D.||
Answer & Explanation

Answer: Option **A**

Explanation:

No answer description available for this question.

54.

What is (void*)0?

A.Representation of NULL pointer
B.Representation of void pointer

C.Error
D.None of above
Answer & Explanation

Answer: Option **A**

55.Can you combine the following two statements into one?

char *p;
p = (char*) malloc(100);
A.char p = *malloc(100);
B.char *p = (char) malloc(100);
C.char *p = (char*)malloc(100);
D.char *p = (char *)(malloc*)(100);
Answer & Explanation

Answer: Option **C**