

Contents

Basics of C	1
Expression	5
If Else & Switch.....	11
Loops.....	18
Functions.....	25
Array & Strings	28
Structure and Union	34
Preprocessor	38
File Handling & Command Line Arguments.....	41

Basics of C

Q. 1 void main()

```
{ printf(“%d”, sizeof(char));
  printf(“%d”, sizeof(int));
  printf(“%d”, sizeof(float));
}
```

a. 1 2 4

b. 1 4 4

c. Error : invalid parameter to sizeof

d. 2 4 4

Q.2 Find invalid rule for an identifier

a. It can start & end with alphabet

b. It can start & end with digit character

c. None of them

Q. 3 Where can I write #include statement?

a. anywhere in function

b. at the start of program

c. anywhere in program, but # should be 1st char. on that line

d. at the start of function

Q. 4

void main()

```
{
  printf(“%d”, printf(“*usm*”) );
}
```

a. Error

b. *usm*5

c. *usm*

d. 5*usm*

Q.5 Statement of C ends with

a. }

b. /

c. \

d. ;

Q.6 Which files contain prototype of standard (library) function?

a. .obj

b. .h

c. .cpp

d. .c

C Programming Question Bank

Q.7 'C' Language is developed by _____.

- a. Stroustrup b. Kernighan Ritchie c. Ken Thompson **d. Dennis Ritchie**

Q.8 Data stored in Computer's memory is in _____.

- a. decimal b. character **c. bin** d. hexadecimal

Q.9 We can not have variables of type _____.

- a. void** b. short c. long double

Q.10

'C' is a _____ level language.

- a. lower **b. high** c. mid Feedback

Q.11 Where can we declare local variables?

- a. None of this **b. start of block** c. start of program

Q.12 When memory is allocated to variables of a program?

- a. when we compile program b. when we save program
c. when we execute program d. when we write program

Q.13 By default local variable contains _____.

- a. 0 b. -ve data **c. garbage** d. !

Q.14 In every C program, first control is received by function _____.

- a. that is declared first in program b. that is defined last in program
c. main d. that is defined first in program

Q.15 The maximum length of an identifier is _____.

- a. 10 char b. 32 char **c. Compiler dependant** d. 8 char

Q.16 Which of the following is a valid identifier?

- a. 26-July **b. 26 July** c. 26_July d. _26July

Q.17 Which of the following statements are true regarding arithmetic operators?

- i) An arithmetic operation between an integer and an integer yields an integer
ii) An arithmetic operation between two floats yields an integer
iii) An arithmetic operation between an integer and a float yields an integer
iv) An arithmetic operation between an integer and a float yields a float
a. i and iv b. i and iii c. ii and iii d. ii and iv

Q.18 We can not have variables of type _____.

- a. long double b. short int **c. void** d. signed char

Q.19 Decimal equivalent of hex 10 is _____.

- a. 16** b. 10 c. 17 d. 12

Q. 20

```
void main( )
{
printf("%c", „%d" );
}
```

- a. garbage **b. d** c. %c d. Error

Q.21

```
void main( )
{ char ch='12';
printf("%d", ch );
}
```

- a. 1 **b. 50** c. 12 d. Error

Q.22 Which of the following are format specifiers?

- i) %f ii) %c iii) %q iv) %z v) %u

Select one:

- a. i, ii, iii & iv b. ii, iii, iv & v **c. i, ii & v** d. i, ii, iv & v

Q.23 Which is invalid escape character?

- a. \\ b. \7 **c. *** d. \"

Q.24 Block of statements are enclosed in

- a. { }** b. < > c. () d. []

Q.25 Which characters are used for multiline commenting?

- a. **/ b. **\ **c. /**/** d. /**\

Q.26 Which is not a translator?

- a. Linker** b. Assembler c. Interpreter d. Compiler

Q.27 Which character is used to precede escape character?

- a. % b. / **c. ** d. none

Q.28 Find odd man out

- a. 97 „a" 0x61 **b. 91 'a' 0x5b** c. 65 „A" 0x41 d. 90 „Z" 0x5A

Q.29

```
void main( )
{
printf("\\65" );
}
```

- a. \5** b. \\65 c. \A d. \65

Q.30 Octal constants are preceded by char _____.

- a. 0 **b. 0(zero)** c. 0x Feedback

Q.31 What getch() does?

C Programming Question Bank

- a. gets char from user & shows it on screen
b. gets char from user & does not show on screen
c. gets char from file
d. shows output to user

Q.32 By default how many decimal places are displayed for %f used for float variable?

- a. 6 b. depends on data c. 2 d. 0

Q.33 Which of the following is not a keyword of C?

- a. void b. register c. n Feedback

Q.34

```
void main( )  
{  
printf(“%d %d %d”, 8 );  
}
```

- a. Error b. 8 8 garbage c. 8 0 0 d. 8 0 garbage

Q.35

```
int x = 10;  
void main( )  
{printf(“%d”,x);  
{int x =20;  
{ int x =30;  
printf(“%d”,x);  
}  
printf(“%d”,x);  
}  
}
```

- a. Erro b. 10 30 30 c. 10 30 20 d. 10 20 30

Q.36 Which is false statement?

- a. 1 is true b. 32767 is true c. 32768 is false d. 0 is false

Q.37

```
void main( )  
{  
printf(“Today\r\bis a\7 great\t\r\\day”);  
}
```

Select one:

- a. day 7 great b. Todayrbis a\7 greattrday c. is a 7 great day d. \day great

Q.38

```
void main( )  
{int ch=26;  
printf(“%X”, ch );  
}
```

C Programming Question Bank

a. 1a

b. 1 A

c. 26

Feedback

Q.39 What is format char for printing data in binary?

a. %B

b. %b

c. none

d. %o

Expression

Q. 1

```
void main( )  
{ int a=2, b, c,d;  
b=c=5;  
d=++a- --b*c%b;  
printf("d", d);  
}
```

a. 3

b. 2

c. -1

d. Error:Lvalue required

Q. 2

```
void main( )  
{ float a=5,b=2;  
printf("%f", a%b*2);  
}
```

a. 2.0

b. 0.5

c. error

d. 1.0

Q.3

```
void main( )  
{ int u1; float u2;  
printf("%d", sizeof(u2+sizeof(u1)));  
}
```

a. 4

b. 8

c. C

d. 6

Q.4

```
void main( )  
{ int i=20, j=30;  
i^=j;  
j^=i;  
i^=j;  
printf("%d, %d", i, j);  
}
```

a. 0,0

b. 30,20

c. 20,30

d. 20,20

Q.5

```
void main( )  
{ int x=32767, y=9;  
x=x+y;  
printf("%d", x);  
}
```

C Programming Question Bank

a. Garbage value b. 32776 c. -32760 d. -32776

Q.6

```
void main( )
{ int p=4;
printf("%d", ++p*p++*++p);
}
```

a. 216 b. none c. 175 Feedback

Q.7 Which is correct statement?

- a. && checks 2nd condition only if 1st condition is true
- b. || checks 2nd condition only if 1st condition is true
- c. && checks 2nd condition only if 1st condition is false

Q.8 Find invalid bitwise operator

a. && b. ~ c. ^ d. >>

Q.9 Which statement will generate an error?

a. (b>c) ? a=b : a=c; b. None of them c. a=b>c ? b : c;

Q.10 Which is not a operator in C?

a. \$ b. ^ c. ~ d. sizeof

Q.11 What is negation of operator <?

a. !< b. ! c. > d. >=

Q. 12

```
void main( )
{ int x=1, y=2, z=6;
x+=x+y*z=8;
printf(" %d", x);
}
```

a. Compi b. 18 c. 14 d. 2

Q.13

```
void main( )
{ char ch=-3;
printf("%d %d %d", -ch, -ch+1, !ch );
}
```

a. 3 132 0 b. -3 132 1 c. 3 -3 1 d. 3 4 0

Question 14

```
void main( )
{ int a=5, b=2;
printf("%d %f %f", a/b, (float)a/b, a/(float)b );
}
```

a. 2 2.5 2.5 b. 2 2.500000 2.500000 c. 2 2.000000 2.500000

Question 15

```
void main( )  
{ char b='C ', c='B';  
printf("%c", c+b-36);  
}
```

- a. A **b. a** c. 133

Q.16 Which unary operator changes operand?

- a. ~ b. ! c. -- d. -

Q.17

```
void main( )  
{ int no 1 =0;  
printf("%d", 5/no 1);  
}
```

- a. Compilation error** b. 32767 c. 5 d. None of them

Q.18

```
void main( )  
{ int p=5, x;  
x=++p*p++;  
printf("%d", x);  
}
```

- a. 36** b. 49 c. 42 d. 30

Q.19

```
void main( )  
{ int xx=3*(5 +1)/(2-2*10);  
printf("%d", xx);  
}
```

- a. -1** b. None of them c. 1 d. 0

Q. 20

```
void main( )  
{ int i=1, j=0, k=i--&&++j || --i;  
printf("%d %d %d", i, j, k);  
}
```

- a. 0 1 0 b. None of these c. 1 1 1 **d. 0 1 1**

Q.21 How to store value of x in y?

- a. x=y; **b. y=x;** c. x= =y;

Q.22

```
void main( )  
{ int i=2, j=3, k=0, m=0 ;  
float a=1, b=2;  
k=i/j *k;
```

```
m=j/i*j;  
a=i*/j*j;  
b=i*j/k;  
printf(“%d %d %f %f ”, k, m, a, b);  
}
```

a. 0 3 0.000000 0.000000

b. Error

c. 0 3 0.000000 2.000000

d. 2 3 0.000000 2.000000

Q.23

```
void main( )  
{ int x=1, y=2, z=3;  
printf(“ %d %d %d”, ++x, x+y, z);  
}
```

Select one:

a. 2 4 3

b.

c. 1 3 3

d. 2 3 3

Q. 24 In the following expression which operation is performed first?

Expression: $a = 3/2 + 9*6/4 - 7 + 2.4/8$

a. 2+9

b. 6/4

c. 3/2

d. 9*6

Q.25

```
void main( )  
{ int s1 = 1, s2=4;  
printf(“%d”, (s1&s2)?5:10 );  
}
```

a. 5

b. 10

c. 0

d. 1

Q.26 Which of the following are unary operators?

a. %(modulus)

b. / (division)

c. =(assignment)

d. -(subtraction)

Q. 27

```
void main( )  
{ int p=10;  
printf(“%d %d %d”, --p, --p, --p);  
}
```

a. 9 8 7

b. 7 8 9

c. 10 9 8

d. 7 7 7

Q.28 Which operator does not work with float?

a. %

b. /

Q.29

```
void main( )  
{ int p=1, q=2, r=3;  
float p=1.0, q=2.0, r=3.0;
```


C Programming Question Bank

```
printf("%d %d %f", p, q, r);  
}
```

- a. error b. 1 2 3 c. 1 2 3.0 d. 1.0 2.0 3.0

Q.30

```
void main( )  
{ int x=2;  
float f1=1.0, f2=2.0, sum;  
sum=x*f1+f2/x+f1*f2;  
printf("%f ", sum);  
}
```

- a. Compilation error b. 1.500000 c. 6.000000 d. 5.000000

Q.31 Which is invalid type of operator?

- a. Relational b. Bitwise c. Logical d. Saving

Q.32 What is not allowed?

- a. ++ k b. +++ k c. + k d. ++++ k

Q.33 Which condition can result into true?

- a. None of them b. $x > y \ \&\& \ x < y$ c. $x \geq y \ \&\& \ y > x$ d. $x != y \ \&\& \ x > y$

Q.34 Which is not a operator in C?

- a. .* b. != c. += d. ->

Q.35

```
void main( )  
{  
printf("%d", 32>>2 );  
}
```

- a. 30 b. 128 c. 8

Q.36 How many ternary operators are available in C?

- a. 1 b. 2 c. 5

Q.37 Which is correct precedence order of binary operators?

(descending order of priority)

- a. Arithmetic, Relational, Logical, Assignment
b. Assignment, Relational, Logical, Arithmetic
c. Arithmetic, Logical, Relational, Assignment
d. Relational, Arithmetic, Logical, Assignment

Q.38 Which type is not available in operators?

- a. unary b. binary c. nnary d. ternary

Q.39

```
void main( )  
{ int v=-5, w=0, x=4;
```

C Programming Question Bank

```
printf("%d", w || v || x&&w);
}
```

- a. 1 b. 5 c. Garbage value is printed d. 0

Q.40

```
void main( )
{ int product;
char ch='CH';
product("%d", product);
}
```

- a. 99 b. 67 c. Garbage d. Error:Character constant too long

Q.41

```
void main( )
{ char ch=20;
printf("%d %d", ch>>4, ch <<2 );
}
```

- a. 180 10 b. 1 80 c. 5 40

Q. 42 Which can not be used as assignment operator?

- a. %= b. ^= c. &= d. !=

Q.43

```
void main( )
{ int p=10;
printf("%d %d %d", p--, ++p, p--); // 10 10 10
-1 9 -1
}
```

- a. 9 11 10 b. 10 9 10 c. 10 10 10 d. error

Q.44 Which operator does not check 2nd condition if 1st condition is false?

- a. & b. || c. && d. >>

Q.45 What is associativity for comma operator?

- a. Right to Left b. Left to Left c. Left to Right d. Right to Right

Q.46

```
void main( )
{ char c1='1', c2='2'; 73+50
int sum=c1+c2;
printf("%c %d", sum, sum);
}
```

- a. c 99 b. 3 garbage c. C 99 d. sign 123

Q.47

```
void main( )
{
```

C Programming Question Bank

```
printf("%d", 5/0 );  
}
```

- a. 32767 b. Runtime error c. **Compilation error** d. 5

If Else & Switch

Q.1 What is the difference between '=' and '==' operators?

- a. = is a assignment operator, whereas == is a logical operator,
b. = assigns value, whereas == compares two quantities
c. – should be used only for assigning integer values to identifiers, whereas == can be used to assign char, float values to identifiers.
d. Both operators mean same when used in condition.

Q.2 Which operator works like if else?

- a. :: **b. ? :** c. -> d. >>

Q.3 Which operators can be used to form a condition?

- a. Relational b. Relational & logical
c. Logical & arithmetic **d. All operators in C**

Q.4 Which operator cannot be used for joining two conditions?

- a. && b. || **c. !** d. Both && and ||

Q. 5 Which operator is used to negate the condition?

- a. - b. ~ **c. !** d. ^

Q. 6 Which is correct statement?

- a. alone if can occur without else** b. alone else can occur without if
c. else should be written before if d. else is compulsory for every if

Q.7 Which nested if-else can be converted to switch?

- a. Every nested if-else can be converted in to switch
b. Having 1 operand common in all conditions & other operand uncommon(contant)
c. aying relational operator == in all conditions
d. both having relational operator == in all conditions and having 1 operand common in all conditions & other operand uncommon(contant)

Q.8 What is possible?

- a. writing switch in if-else b. writing if-else in switch
c. writing one switch another switch **d. All of the remaining options**

Q.9 What is not required for writing switch?

- a. default b. case c. break **d. continue**

Q.10 What is true about default?

- a. default can be written anywhere in switch**

C Programming Question Bank

- b. default must be last case in switch
- c. default must be first case in switch
- d. default is compulsory in switch

Q. 11 What is most false about switch?

- a. **All cases require continue**
- b. All cases require break
- c. Last case doesn't require break
- d. First case may not have break

Q.12 Switch can not handle _____ data type

- a. char
- b. int
- c. float
- d. **float & double**

Q.13 What is false about switch cases?

- a. cases have to be constant expression
- b. cases can be fractional
- c. cases can be strings
- d. **Both cases can be fractional and cases can be strings**

Q.14 What is true about switch cases?

- a. more than 1 cases having same action can be written together separated by comma
- b. more than 1 cases can be joined by using &&, ||
- c. **duplicate cases are not allowed**
- d. small case & upper case are treated as same case

Q.15 switch is preferred over nested if-else because

- a. It increases complexity of a program
- b. It improves execution speed of a program
- c. **It increases readability of a program**
- d. It reduces size of compiled code

Q.16 Which is keyword of C?

- a. Case
- b. Default
- c. If
- d. **None of these**

Q.17 Which character is used in switch to separate case from its action part

- a. blank
- b. **:**
- c. ;
- d. ,

Q.18 Which condition is correct to check whether ch is a capital letter?

- a. `ch>='A' || ch>='Z'`
- b. `ch<='A' && ch<='Z'`
- c. `ch>='A' || ch<='Z'`
- d. **`ch>='A' && ch<='Z'`**

Q.19

```
void main( )
```

```
{ int m=10, n=15;
```

```
if(!(m)&& m) printf("%d", m++);
```

```
else printf("%d", n++);
```

```
}
```

- a. 15 **b. 10** c. 16 d. 11

Q.20

```
void main( )  
{ float f=5.4;  
    if(f==5.4) printf("equal");  
    else      printf("not equal");  
              printf("%f", f);  
}
```

- a. equal b. not equal 5.400000 c. not equal **d. 5.400000**

Q.21

```
void main( )  
{ int a=2, b=3;  
    if(a=b); printf("equal");  
    else printf("not equal");  
}
```

- a. not equal b. equal c. No output **d. Error**

Q.22

```
void main( )  
{ if( ! 123.4 ) printf("WELCOME");  
    Else printf("GET OUT");  
}
```

- a. WELCOME **b. GET OUT** c. no output d. Error: improper if condition

Q.23

```
void main( )  
{ static int i;  
    if(!(i=1)) printf("USM");  
    else printf("usm");  
}
```

- a. usm
b. USM
c. Error:static variable not allowed in C
d. Error: invalid if condition

Q.24

```
void main( )  
{ int a=2,b=3;  
    if(a>b);  
    printf("*");  
    printf("+");  
}
```

C Programming Question Bank

- a. * b. + c. *+ d. no output

Q.25

```
void main( )
{
    int a=2,b=5,c=1;
    if(a>b)
        if(a>c) printf("*");
    else printf("*");
    printf("+");
}
```

- a. *+ b. **+ c. *** d. +

Q.26

```
void main( )
{
    int a=3,b=0;
    if(a,b) printf("*");
    else printf("+");
}
```

- a. * b. + c. *+ d. Error

Q.27

```
void main( )
{
    int k=1,b=0,c=++b;

    if( k-- >0 ) printf("*");

    else printf("+");

    print("%d", k);
}
```

- a. *0 b. +0 c. *1 d. +1

Q.28

```
. void main( )
{
    int a=0,b=1;

    if(++a=---b) printf("%d %d", --a, b++);

    else printf("%d %d", a--, ++b);
}
```

- a. 0 1 b. 0 0 c. -1 0 d. 1 1

Q. 29

```
void main( )
```

```
{ int a=3,b=1;
```

```
    if( b>a, a>b) printf("%d",--b);
```

```
    else printf("%d", a++);
```

```
}
```

a. 3

b. 1

c. 0

d. Error

Q. 30

```
void main( )
```

```
{ int a=0,b=2;
```

```
    if(a++, b--, --b) printf("%d %d", a, b);
```

```
    else printf("%d %d", b, a);
```

```
}
```

a. 0 1

b. 2 1

c. 1 0

d. Error

Q.31

```
if(a>b) b=a;
```

```
if(c>b) b=c;
```

```
printf("%d", b);
```

What is represented by above code?

a. it prints value of b as it is

b. it prints minimum of 3 nos. a, b, c

c. it prints maximum of 3 nos. a, b, c

d. it prints average of 3 nos. a, b, c

Q.32

```
void main( )
```

```
{
```

```
    int m=0,n=5/2+3*6;
```

```
    if( !m++ && ++m) printf("%d", m++);
```

```
    else printf("%d", m--);
```

```
}
```

Select one:

a. 1

b. 2

c. 3

d. 0

Q.33

```
void main( )
```

```
{
```

```
    int m=3, n=4;
```

```
    if(m&n) printf("%d", m--);
```

```
else printf("%d", n++);  
}
```

- a. 3 b. 4 c. 2 d. 5

Q.34

```
void main( )  
{  
    int m=0, n=-1;  
    if(~m && ~n) printf("%d", n);  
    else printf("+%d", ~m);  
}
```

- a. +-1 b. -1 c. +32767 d. 32768

Q. 35

```
void main( )  
{  
    char a='A';  
  
    if(a== 'a' || (a=='B') || a) a=a+32;  
  
    else a='c';  
  
    printf("%c", a);  
}
```

- a. a b. b c. c d. Error

Q.36

```
void main( )  
{ char a='A';  
  
    if(a== 'a' || a== 'B' || 'b') a+=32;  
  
    else a='c';  
  
    printf("%c", a);  
}
```

- a. a b. c c. A d. Error

Q.37

```
void main( )  
  
{ char a='A';  
  
    if(a== 'a' || a=='B' || 'b') a+= ' ';
```



```
else a= 'C'+ ' ';
```

```
printf("%c", a);
```

```
}
```

a. a

b. c

c. A

d. Error

Q.38

```
void main( )
```

```
{ int a=2,b=3;
```

```
if(a<=b)
```

```
if(a=b) printf("***");;
```

```
else printf("*****");
```

```
}
```

a. **

b. ****

c. No output

Q.39

```
void main( )
```

```
{ float j=0.1;
```

```
switch( j )
```

```
{case 0.1 : printf("%f", j++); continue;
```

```
case 0.2 : printf("%f", j ); break;
```

```
default : printf("Default");
```

```
}
```

```
}
```

a. 0.1

b. 0.1 0.2

c. 0.1 Default

d. Error

Q. 40

```
void main( )
```

```
{char c='a';
```

```
lable: switch (c)
```

C Programming Question Bank

```
{ case 'a': ++c; goto lable;

    case 'b': printf("%c", c++);

    case 'c': printf("%c", c);

}
```

a. b

b. c

c. b c

d. a b c

Loops

Q.1 What is not a difference between while & do-while loop?

- a. In while loop condition is checked 1st & in do while it is checked last
- b. do-while loop executes at least once whereas while loop may not execute in worst case
- c. while loop stops when condition is false whereas do-while stops when condition is true**
- d. In while loop, condition decides execution of current iteration whereas in do while, it decides execution of next iteration.

Q. 2 Who allows control to flow inside or outside loop?

- a. loop statements
- b. loop expression
- c. loop condition**
- d. initialization

Q.3

```
void main( )
{ int i=0;
  for(;i=2;){ printf("%d ", i); i++; }
}
```

a. 0

b. 0 1 2

c. 0 1

d. Infinite loop

Q.4 In for(...) how many semicolons are allowed?

- a. <=2
- b. exactly 2**
- c. >=2
- d. >=0

Q. 5

```
void main( )
{ char ch='\0';
  while(ch)
  { printf(" end of output is %c", ch);
    break;
  }
  printf(" null character");
}
```

a. null character

b. end of output is 0 null character

c. end of output is \0 null character

d. end of output is null character

Q.6

```
void main( )
{ int i=0;
```

C Programming Question Bank

```
for(;i<=5;) printf("%d ", i=i*2);  
}
```

- a. 0 2 4 b. 0 1 2 3 4 5 Incorrect **c. 0 is printed infinitely** d. Error

Q.7

```
void main( )  
{ int sum=0,n=8;  
do  
{ sum=n%2;  
n/=2;  
printf("%d", sum);  
}while(n);  
}
```

- a. 0000 b. 1000 **c. 0001** d. 000

Q. 8

```
void main( )  
{ int i=0;  
for(;i<=3; printf("%d ", ++i) );  
}
```

- a. 1 2 3 **b. 1 2 3 4** c. 0 1 2 3 d. Error

Q.9

```
while(k<=5) printf("%d", k++);  
If k contains initially -5, then how many times above loop will execute?
```

- a. 0 b. 10 **c. 11** d. 9

Q.10 Which logic will give output :

1
1 2
1 2 3
1 2 3 4

- a. for(a=1; a<=4; a++)
{ printf("\n");
for(b=1; b<=a; b++) printf("%2d", a);
}
**b. for(a=1; a<=4; a++)
{ printf("\n");
for(b=1; b<=a; b++) printf("%2d", b);
}**
c. for(a=1; a<=4; a++)
{ printf("\n");
for(b=0; b< 4; b++) printf("%2d", a);
}
d. for(a=0; a < 4; a++)
{ printf("\n");
for(b=1; b<=4; b++) printf("%2d", b);

```
}
```

Q.11

```
void main( )  
{ int k=3;  
while(k)  
{int k=1;  
printf("%d ", k);  
k--;  
}  
k--;  
}
```

- a. 1 b. 3 2 1 **c. Infinite loop** d. Error

Question 12

```
void main( )  
{ int x=3;  
do  
{printf("%d ", x--);  
}while(x&&0);  
}
```

- a. 3** b. 3 2 1 c. 2 1 d. none

Q.13

```
void main( )  
{ int i=0,j=-1;  
for(i=1; i<=3; i++)  
{ for(j = 1; j<=3; j++)  
{ if(i==j) break;  
printf("%d%d",i, j);  
}  
}  
}
```

- a. 121321233132 **b. 213132** c. 111213212223313233 d. No output

Q.14

```
void main( )  
{ int i, j;  
for(i=1; i<=3; i=i+2)  
for(j=1; j<=3; j=j+2)  
printf("%d", i+j);  
}
```

- a. 2 4 4 6** b. 1 2 2 3 c. Infinite loop d. No output

Q.15

```
void main( )  
{ char ch='\0';
```

C Programming Question Bank

```
while( ch<=127) { printf("%c %d ", ch++, ch ); }  
}
```

- a. Prints 127 ASCII char b. Prints entire ASCII table
c. Infinite loop d. No output

Q.16 In worst case, how many times will do – while loop executes (minimum)?

- a. 0 b. 1 c. n times d. infinite

Q.17

```
void main( )  
{ int k=3;  
while(k)  
{ int K=1;  
printf("%d", k);  
k--;  
}  
}
```

- a. 1 b. 3 2 1 c. Infinite loop d. Error

Q.18

```
void main( )  
{ int no=123;  
while(no)  
{ no/=10;  
printf("%d", no%10);  
}  
}
```

- a. 2 b. 210 c. 21 d. 213

Q.19 Which part of loop executes only once?

- a. loop statements b. loop expression
c. loop condition d. initialization

Q.20

```
void main( )  
{ int i=1;  
while(i<=32767)  
{ printf("%d ", i);  
i++;  
}  
}
```

- a. 1 2.....32766 b. 1 2.....32767 c. 1 2.....32768 d. Infinite loop

Q.21

```
void main( )  
{ int i=0j=1;  
for(i=1,j=0; j<i; i++, j++)
```

C Programming Question Bank

```
{ if(i= =j) goto label2;  
else printf("%d", i);  
}  
label2: printf("%d", j);  
}
```

- a. 1 **b. 1....3276732767** c. 1....32767 d. Infinite loop

Q.22 Missing condition in for loop will generate_____.

- a. Compiler error b. Runtime error c. Warning **d. Infinite loop**

Q. 23 What continue does?

- a. It helps to continue execution of a program
b. It skips that iteration & continues loop
c. It continues that iteration & then takes control outside loop
d. It continues execution of a function by skipping a loop

Q.24 The statements which are repeated in loop are called

- a. loop statements** b. loop expression c. loop condition d. initialization

Q.25

```
void main( )  
{ int i=0;  
for(i=1;;i++)  
if(i= =5)  
{ printf("%d", i);  
break;  
}  
else  
{ printf("%d",i++);  
continue;  
}  
}
```

- a. 12345 b. 246 **c. 135** d. Error

Q. 26 Missing condition in while loop will generate_____.

- a. Compiler error** b. Runtime error c. Warning d. Infinite loop

Q.27 In do while loop, 5th time condition is checked & now resulted into false, so how many times loop has executed?

- a. 4 **b. 5** c. 6 d. 0

Q.28 Which loop will not output nos. 1 to 10

- a. for(a=0; a<10; a++) printf("%d",a+1); b. for(a=0; a<10;) printf("%d", ++a);
c. for(a=0; a++<10;) printf("%d", a); **d. for(a=0; a<10; a+1) printf("%d", a);**

Q. 29 What is correct execution sequence in a for loop?

- a. loop statements, loop expression, loop condition, initialization

C Programming Question Bank

- b. loop expression, loop statements, loop condition
- c. loop statements, loop condition, loop expression
- d. loop condition, loop statements, loop expression**

Q.30 In which loop, condition decides execution of next iteration?

- a. while
- b. do while**
- c. for
- d. none

Q.31

```
void main( )
```

```
{
```

```
While (1) printf("USM C");
```

```
}
```

- a. condition in while loop is illegal

c. while loop is an infinite loop

- b. 'USM C' will be printed once

- d. Linker error is generated

Q.32 What is not a loop in C?

- a. for
- b. While
- c. repeat until**
- d. do while

Q.33 Which statement helps to stop the loop?

- a. loop statements
- b. loop expression**
- c. continue
- d. initialization

Q.34

```
void main( )
```

```
{
```

```
int j=1;
```

```
for(;j<=3; j++) printf("%d", j);
```

```
}
```

- a. 0

b. 123

- c. Infinite loop

- d. No output

Q.35

```
void main( )
```

```
{ int i=2j=3;
```

```
do
```

```
{ if(j= =i)
```

```
{ printf("less"); break;
```

```
}
```

```
j+=i;
```

```
printf("more");
```

```
}while(i!=j-3);
```

```
}
```

- a. less

b. more

- c. 'more' is printed thrice

- d. Infinite loop

Q.36 In while loop, 7th time condition is checked & now resulted into false, so how many iterations executed?

- a. 0
- b. 5
- c. 6**
- d. 7

Q.37

C Programming Question Bank

```
void main( )
{ int j=1;
  while( j<=255)
  { printf("%c %d ", j, j );
    j++;
  }
}
```

- a. Prints garbage **b. Prints ASCII table** c. Infinite loop d. No output

Q. 38 Which loop does not require to write ';' after loop condition?

- a. while** b. do while c. for d. none

Q.39

```
void main( )
{
  char ch=49 - '1';
  while(++ch) putchar(ch);
}
```

- a. Prints all ASCII chars except last **b. Prints all ASCII char except First**
c. Infinite loop d. No output

Q. 40 Which statement can stop the loop?

- a. continue **b. break** c. initialization d. if

Q.41

```
void main( )
{
  int i=0;
  while(++i<5) printf("%d", ++i);
}
```

- a. 24** b. 1234 c. 246 d. Error

Q. 42

```
void main( )
{
  while(4<5) printf("hi!");
}
```

- a. hi! **b. hi! Printed infinitely** c. No output d. Error

Q. 43

```
void main( )
{
  int i=0;
  for(i=1;i<=4,i++)
  { if(i==2) continue;
    else
    { printf("%d", i);

```



```
break;  
}  
}  
}
```

- a. 1 3 4 b. 1 2 3 c. 1 d. 2

Q. 44 Which loop allows to omit condition?

- a. while b. do while c. for d. none

Q. 45 In for loop, more than one initializations or loop expressions are separated using_____.

- a. ; b. , c. blank d. tab

Q.46

```
void main( )
```

```
{  
int i=10;  
for(;;) i++ printf("hello!");  
}
```

- a. hello! b. 'hello!' is printed 10 times c. Infinite loop d. Error

Q. 47 In worst case, how many times while loop executes (minimum)?

- a. 0 b. 1 c. n times d. infinite

Q. 48 What is not used to control loop iterations?

- a. loop statements b. loop expression c. loop condition d. initialization

Q.49

Which loops are similar in behavior?

- a. for & do while b. while & do while c. while & for d. do while, for & while

Q. 50

```
void main( )
```

```
{  
int i=0, j=0;  
while(j= =i)  
{ printf("%d %d ", i, j);  
i++; j--;  
}  
}
```

- a. 1 1 b. 0 1 0 1 0 1 c. 0 0 d. Infinite loop

Functions

Q.1 What is true regarding a return statement in a function?

Select one:

- a. There cannot be more than one return statements in a function
b. A return statement must be written at the end of the function

C Programming Question Bank

c. A function may not have return statement

d. All the above are true

Q.2 How many times & when memory is allocated to formal parameters?

a. Only once during compilation time

b. Only once during execution time, at the start of program

c. Only once during executing time, before function call

d. For every call of that function

Q.3 Which following statement is false about parameters?

a. Formal Parameters should be same as actual parameters

b. Actual Parameters should be same as formal parameters

c. Number of Actual & formal parameters should be same

d. Data type sequence of actual parameters should be same as formals

Q. 4 The declaration "int *P(char a[]);" indicates

a. P is pointer to integer

b. P is pointer to function that returns an integer

c. P is function that returns an integer pointer

d. Illegal declaration of P

Q.5 What is true about returning value from a function?

a. We must collect a value when function has return statement

b. Even though function is returning a value, we may not collect

c. We can collect a value even though function has void returning data type

d. We can not collect a value when function has no returning data type

Q. 6 Which two names cannot be same?

a. Formal parameter & local variable

b. Formal parameter & actual parameter

c. Formal parameter & function name

d. Local variable & function name

Q. 7 For every function call in a program, compiler require_____.

a. its definition

b. its declaration

c. both, its declaration & definition

d. nothing

Q. 8 _____requires function definition.

a. Linker

b. Loader

c. Compiler

d. Editio

Q. 9 How many values can be returned from a function using return statement?

a. only 1

b. maximum 2

c. As many we want, by writing more than one return statements

d. As many we want, by using comma operator

Q.10 Which statement is true regarding function?

a. Any C program contains at least one function

b. A function can be defined inside another function

c. In a C program there can be more than one functions with the same name

C Programming Question Bank

d. A C program can be written without functions

Q.11 _____ function executes on its own.

- a. All **b. No** c. main d. All standard

Q.12 What is false about function main?

- a. main can call any other function b. any other function can call main
c. main can call main **d. No function can call main**

Q.13 During execution, parameters passed to the program are received by function_____.

- a. that is defined as main** b. that is defined 1 c. that is declared 1

Q.14 The declaration "void fun(int);" indicates that the function

- a. returns a float value b. has no arguments
c. returns nothing d. has default arguments

Q.15 Parameters passed in a function call are called _____.

- a. Formal Parameters b. Receiving Parameters
c. Local Parameters **d. Actual Parameters**

Q.16 What is true about function declaration?

- a. Function definition includes its declaration**
b. Function declaration includes its definition
c. Function call includes both its declaration & definition
d. Function declaration includes its call

Q.17 Which of the following statement is true regarding recursive function?

- a) It is also called as a circular definition
b) It occurs when a function calls another function more than once
c) It occurs when a statement within function calls itself
d) A recursive function can not have a return statement within it

- a. a & c** b. a & b c. b & d d. a, c & d

Q.18 Which statement is true?

- a. Function is a block of statements that can be executed number of times
b. Function can have loop inside & same function can be called in loop
c. Call by reference is same as passing address to function
d. All above statements are true

Q.19 What is true regarding a returning data type of a function?

- a. Function without returning data type can not return a value
b. Function having returning data type void can return a value
c. Function having returning data type void can not have return statement
d. 1, 2 & 3 are false

Q.20 What happens in call by value?

C Programming Question Bank

- a. Changes made in formal Parameters are reflected in actual parameters
- b. Changes made in formal Parameters are not reflected back in actual parameters**
- c. Addresses of actual parameters are passed to formal parameters
- d. Values of actual parameters are passed to local variables of a called function

Q.21 What is true about multidimensional array as formal parameter?

- a. We can pass single dimensional array to a Function but not multidimensional
- b. We can pass multidimensional array, but size of all dimensions should be same
- c. We can pass multidimensional array, but size of all dimensions should be constant
- d. Size of 1st dimension can be empty, but we need to mention sizes of remaining dimensions**

Q.22 A function has by default returning data type_____.

- a. void
- b. char
- c. int**
- d. none of these

Q.23 What is false about function main?

- a. main can have parameters
- b. main can return a value
- c. main can be called
- d. variables defined inside main become global variables**

Q.24 What we can have in function name?

- a. blank
- b. underscore**
- c. special char
- d. #

Q.25 In recursive functions, how many times memory is allocated to its local variables?

- a. Only once before that function is called
- b. Depends on no. of recursive calls at runtime**
- c. Depends on how many times function is called in a program
- d. Only once during compilation time

Array & Strings

Q.1 void main()

```
{ int a[ ]={10,20,30,40,50}, j;  
  
    for(j=0; j<5; j++) printf("%d",*a);  
  
}
```

- a. Error : L value required
- b. 10 20 30 40 50
- c. Garbage value printed 5 times
- d. 10 10 10 10 10**

Q.2 What is true about size of an array?

- a. All remaining options are true**
- b. Size can be omitted if array is initialized
- c. Size of array must be fixed (constant)
- d. There is no limit on size of array

Q.3

void main()

```
{ int ary[2];  
  
    ary[-2]=2;  
  
    printf("%d", ary[2-4] );  
  
}
```

- a. 2-4 b. garbage c. Error **d. 2**

Q. 4

```
int num[5];  
    Num[5]
```

What is true about above two statements?

- a. Both statements indicate array size
b. 1st statement indicates array size & 2nd indicates element
c. Both statements indicate element of an array
d. 1st statement indicates element & 2nd indicates array size

Q.5 Which is not a standard function used for string comparison?

- a. strncmpi** b. stricmp c. strncmp d. strnicmp

Q. 6 Which function can be used to find last occurrence of a character in a string?

- a. strchr b. strlchr **c. strrevchr** d. strrchr

Q.7 Which of the following statements are correct regarding arrays :

- i) Array is a collection of similar type of elements
ii) Array is a group of same type of variables
iii) Array has sequential storage
a. i and iii b. iii only **c. All are true** d. ii and iii

Q.8

```
void main( )  
{  
    char dac[ ];  
    dac[0]=65;  
    printf("%c", dac[0] );  
}
```

- a. 65 b. A **c. Error** d. a

Q.9 Indexing of array starts from

- a. depends on compiler b. size-1 c. 1 **d. 0**

Q.10

```
void main( )  
{  
    int a[ ]={20,30,40,50,60}, *j=a;  
    j+=3;
```

C Programming Question Bank

```
printf("%d",*j);
}
```

- a. 50 b. 20 c. Compilation Error d. 40

Q.11 A[j] is not same as _____.

- a. *(A+j) **b. A+j** c. *(j+A) d. j[A]

Q.12 There can not be array of _____.

- a. pointer b. structure c. array **d. void**

Q.13

```
char *fun( )
{
char result[100]="Hello world";
    strcpy(result, "Anything is Good");
    return result;
}
void main( )
{
```

```
    char *j;
    j=fun( );
    printf("%s", j);
}
```

- a. Compilation Error b. Hello world c. Garbage **d. Anything is Good**

Q.14

```
void main( )
{
    char a[6*3/2]={'a', 'b', 'r', 'd', 'c'};
    printf("%c", a[3] );
}
```

- a. c b. Compilation Error **c. d** d. r

Q. 15

```
void main( )
{
    char p[5]={0,0,65,66};
    printf("%d %d", p[1], p[4]);
}
```

- a. 0 66 b. 0 garbage c. Error **d. 0 0**

Q. 16

```
void main( )
{ int a[ ]={10,20,30,40,50}, i,*k;
    k=&a[3]-3;
for (i=0;i<=4;i++)
{ printf("%d", *k);                      k++;
```

C Programming Question Bank

```
} }
```

a. 10 20 30

b. Error

c. 10 20 30 40 50

d. 10 20 30 40

Q.17

```
void main( )
```

```
{
```

```
    int M[12]; M[0]=-12; M[11]=-12;
    printf("%d %d %d", *(M+0), *(M+11), *M );
```

```
}
```

a. -12 -12 12

b. 0 11 garbage

c. -12 12 -12

d. -12 -12 -12

Q.18 Which is not a standard string function in C?

a. strtok

b. **strno**

c. strset

d. strrchr

Q.19

```
void main( )
```

```
{
```

```
    int i=10;
    char ch='F', str[10]=" Born To Code";
    sprintf(str,"%d %c",i, ch);
    printf("%s",str);
```

```
}
```

a. 10 F

b. 10 F BornToCode

c. **Error**

d. Born To Code

Q.20

```
void main( )
```

```
{ char names[]={“Aditi”, “Satish”, “Dipti”, “Sohan”, “Mohan”};
```

```
    char *t;    int i;
```

```
    t=names[3];
```

```
    names[3]=names[4];
```

```
    names[4]=t;
```

```
    for(i=0;i<=4;i++) printf(“%s” names[i]);
```

```
}
```

a. Aditi Satish Dipti Mohan Sohan

b. Aditi Satish Dipti Mohan Mohan

c. Aditi Satish Sohan Dipti Mohan

d. **Error**

Q.21 To store string, array of _____ is required

a. string

b. int

c. void

d. **char**

Q.22 Which is the following statement are correct regarding arrays?

i) Array elements are stored in contiguous memory locations

ii) Size of array can be mentioned anywhere in the program

iii) The Expression arr[i] refers to i+1 element in array arr

a. Nothing is true

b. i only

c. All are true

d. **i and iii**

Q. 23

```
void main( )
```

C Programming Question Bank

```
{ char a[12]={‘U’,‘s’,‘m’, ‘C’,‘o’,‘m’,‘p’,‘u’,‘t’,‘e’,‘r’};
    printf(“%s”, a);
}
```

- a. **Usm Computer garbage** b. Error c. Usm d. Usm Computer

Q.24 If we print ‘\0’, what is seen?

- a. \0 **b. blank** c. \\0 d. Error : can not print \0

Q.25

```
void main( )
```

```
{
    char *dest,*source=“USM ITS”;
    dest=malloc(strlen(source));
    strcpy(dest, source);
    printf(“%s”, dest);
    free(dest);
}
```

- a. free() fails **b. USM ITS** c. Compilation error d. strcpy() fails to copy string

Q.26 What is maximum size possible for array of char?

- a. 128 b. 256 c. no limit (entire memory) **d. 65535**

Q. 27

```
void main( )
```

```
{
    int n[3][3]={2,4,6,8,5,3,5,1}, *ptr ,i;
    Ptr=&n[1][0];
    for(i=0;i<=3;i++) printf(“%d”, *(ptr+i));
}
```

- a. 3 6 8 5 **b. 8 5 3 5** c. 6 3 garbage garbage d. 5 3 5 1

Q.28 Which header file should be included for using standard string functions?

- a. string.c b. stdlib.h **c. string.h** d. stdio.h

Q.29 Array can be passed to function by passing its_____.

- a. None of the remaining options b. address **c. values & address** d. values

Q.30 What is used to access individual element of an array?

- a. () **b. []** c. -> d. { }

Q. 31 String ends with

- a. \\0 b. ‘0’ c. NULL **d. ‘\0’**

Q. 32 What can not be used for initializing array?

- a. ' ' b. { } **c. []** d. “ ”

Q.33 What is true about array name?

C Programming Question Bank

- a. Array name is a variable pointer storing starting address of an array
- b. Array name is a constant pointer storing addresses of all array locations
- c. Array name is not a pointer
- d. Array name is a constant pointer storing starting address of an array**

Q.34 What is minimum size possible for an array?

- a. 0
- b. none
- c. 2
- d. 1**

Q.35 Arrays can be initialized provided they are _____.

- a. extern
- b. All are the remaining options**
- c. static
- d. auto

Q.36 _____ dimensional array is not possible.

- a. 0**
- b. 2
- c. 3
- d. 6

Q.37 String constant is enclosed in

- a. " "**
- b. * *\
- c. ' '
- d. { }

Q.38

```
void main( )
```

```
{ int i, fun1( ), fun2( ), fun3( );
```

```
    int (*f[3])( );
```

```
    f[0]=fun1; f[1]=fun2; f[2]=fun3;
```

```
    for(i=0;i<3;i++) (*f[i]) ( );
```

```
}
```

```
fun1( ) { printf("USM");      fun2( ); }
```

```
fun2( ) { printf("COMPUTER"); }
```

```
fun3( ) { printf("EDUCATION"); }
```

a. USM COMPUTER EDUCATION

b. USM COMPUTERCOMPUTEREDUCATION

c. USM COMPUTER

d. Error

Q.39

```
void main( )
```

```
{ char usm[1]={ '2'};
```

```
    usm++;
```

```
    printf("%d", usm );
```

```
}
```

- a. 3
- b. Error**
- c. garbage
- d. 51

Q.40

```
void main( )
```

```
{ int s[2][ ]={3,4,5,6,7,8,9,10};
```

```
    printf("%d", s[1][2] );
```

```
}
```

- a. 8
- b. Error**
- c. 6
- d. 9

C Programming Question Bank

Structure and Union

Q.1 Which storage class can be used to speed up process?

- a. auto b. static c. extern **d. register**

Q.2 Which storage class is used to increase scope of a variable?

- a. auto b. static **c. extern** d. register

Q.3 How many storage classes can be used for a variable at a time?

- a. 0 **b. 1** c. 2 d. 4

Q.4 Which is not a register?

- a. AX b. SI **c. ZX** d. DS

Q.5 _____ storage class variable can have one definition & multiple declarations?

- a. auto b. static **c. extern** d. register

Q.6 Which of the following variable do not have default initial value Zero?

- a. global b. extern c. static **d. none of the above**

Q.7 Consider following declaration Struct DAC {int j; union { double ACTS; int p; }mum; }*USM; How to access ACTS through USM?

- a. USM->mum->ACTS **b. USM->mum.ACTS**
c. USM.mum->ACTS d. USM.mum.ACTS

Q. 8 void main() { long n=2.5; int m=3; long int p=m+n; printf("%d", p);

- a. 5** b. 5.5 c. 5.0 d. Compilation error

Q.9

```
struct s
{ char ch; int j;
};
display (struct s *x1)
{
printf("%d %c", x1->j, x1->ch);
}
void main( )
{
static struct s s1 ={'S',1}, s2={'R',4}; s2=s1; display(&s2);
}
```

- a. 1 S** b. 4 R c. Error: "s2=s1" d. Error: "x1->j,x1->ch"

Q.10

```
struct PP
{ int k; char c; };
QP(struct PP a);
void main( )
{
struct PP n= {110,65}; QP(n); } QP(struct PP a) { printf("%d ..%c" a.k, a.c);
```

}
a. 110...65 **b. 110...A** c. Error d. 65...65

Q.11 void main()
{ struct num1 { char ch[7]; char *s; };
struct num2 { char *c; struct num1 s1; };
static struct num2 s2 ={"apple", "orange", "lemon"};
printf("%s %s", s2.c,s2.s1.s);
printf("%s %s", s2.c++, ++s2.s1.s);
}
a. apple lemon orange b. apple lemon apple orange
c. apple lemon pple emon **d. apple lemon apple emon**

Q.12
void main()
{ stuct s1 { char *z; int l; struct s1 *p; };
static struct s1 a[] = {{"Kerala", 6, a+2},{ "Punjab",5,a+1}, {"Delhi",4,a}};
struct s1 *ptr=a+1;
printf("\n%s%s%s", a[0].z, ptr->z, a[2].p->z);
}
a. Kerala Kerala Kerala b. Kerala Punjab Delhi
c. Kerala Punjab Punjab **d. Kerala Punjab Kerala**

Q.13
void main()
{
 auto i=5.5; register n=5.5;
 Printf("%d %d", i, n);
}
a. 5.5 5.5 b. 5 5.5 c. 5.5 5 **d. 5 5**

Q. 14
int check1(int a)
{
 return(++a);
}

int check2(int a)
{return(a++);
}

void main()
{
 static int x=1;
 printf("%d", check1(x));
 printf("%d", check2(x));
}
a. 22 b. 23 c. 12 **d. 21**

Q.15

```
void main( )  
{  
    double long x=0;  
    printf("%lf", x);  
}
```

- a. **0.000000** b. 0 c. Illegal datatype d. Illegal Format specifier

Q.16

```
void main( )  
{  
    register double sum;  
    printf("%d", sizeof(sum));  
}
```

- a. 4 b. 2 **c. 8** d. Invalid datatype

Q.17

```
static int x=3, y, z;  
void main( )  
{  
    int a; a=x++ * ++y +++z;  
    printf("%d", a);  
}
```

- a. 3 **b. 4** c. 5 d. Garbage

Q. 18

```
void main( )  
{  
    signed char c=`11';  
    printf("%d %c", c, c);  
}
```

- a. 59 11 **b. 49 1** c. 49 11 d. 49 – 11

Q.19

```
int FUN=100  
main( )  
{  
    extern int FUN=200;  
    printf("%d", x);  
}
```

- a. 100
b. 200
c. FUN should be of datatype: 'int extern'
d. extern variable can not be initialized

Q.20

```
static char ch=`a';
```

```
void main( )
{
    extern char ch;
    { ch='b';
    }
    printf("%c", ch);
}
```

- a. a **b. b** c. Garbage d. Compilation error

Q.21

```
int fin;

int init;
int call1(int c) {return(c++); }
int call2(int c) {return(++c); }
void main( )
{
```

```
    register int fin;
    printf("%d", init);
    fin=call1(init);
    printf("%d", fin);
    fin=call2(init);
    printf("%d", fin);
}
```

- a. 001** b. 011 c. 012 d. Error

Q. 22

```
static int k=15,
m=20; change( )
{
    printf("%d %d", k, m);
}
```

```
void main( )
{
    static int k=2, m=1;
    printf("%d %d", k, m);
    change( );
}
```

- a. 2 1 2 1 **b. 2 115 20** c. 15 20 15 20 d. 2 10 0

Q.23

```
auto int ai;
void main( )
{
    ai=ai+10;
    printf("%d", ai);
}
```

- a. 10** b. Garbage c. 0 d. Compilation error

Q.24

```
add( )
{
    static char ch='n';
    printf("%c", ch);
    ch=ch-2;
}

void main( )
{
    add( );
    add( );
    add( );
}
```

a. n n n b. j 1 n **c. n 1 j** d. n 1 k

Q.25

```
enum records
{
    one, two=4, three, four=4};

void main( )
{
    printf("%d,%d,%d,%d", one, two, three, four);
}
```

a. 0,4,5,4 b. 1,4,3,4 c. 1,4,5,4 d. Error

Q.26

```
void main( )
{
    enum days {mon,tue,wed,sat,USM};
    enum days d1,d2,d3; d1=mon+USM;
    d2=wed-sat; d3=USM*tue;
    printf("%d %d %d",d1,d2,d3);
}
```

a. 4 -1 4 b. 1 5 5 c. 0 4 4 d. Compilation error

Q. 27 Which operator cannot be used to access structure member?

a. . **b. ::** c. -> d. 2 & 3

Preprocessor

Q.1 Which header file is not available in standard library?

a. string.h b. ctype.h **c. matrix.h** d. time.h

Q. 2 Which is not a preprocessor directive?

a. include b. undef c. pragma **d. elifdef**

C Programming Question Bank

Q.3 #define SQUARE(x)=(x*x); #define csi(m,n) (SQUARE(m)>SQUARE(n) ? n : m) void main() { int s; s=csi(2,-3); printf("%d",s); }

- a. 2 b. -3 c. 4 d. error

Q.4 #define CASE (a>=65&&a<=90) void main() { char b='Q'; if(CASE) printf("Upper case"); else printf("Lower case"); }

- a. Lower case b. Upper case c. **Compilation error** d. None of these

Q.5 Macro can be disabled using preprocessor directive_____.

- a. undefine b. **undef** c. dontdef d. undefined

Q.6 #define HEADERFILE "conia.h" #include HEADERFILE void main() { window(10, 10,40,20); cprintf("USM COMPUTER EDUCATION"); cprintf("A PERFECT WAY TO IT"); }

- a. USM COMPUTER EDUCATIONA PERFECT WAY TO IT
b. USM COMPUTER EDUCATION
c. **USM COMPUTER EDUCATION A PERFECT WAY TO IT**
d. Compilation error

Q. 7 What does a preprocessor?

- a. Compiles a program
b. It loads a program into memory for execution
c. It links all the required modules and files to the program
d. **It provides an expand source code to the compiler**

Q. 8 # define square(x) (x*x) void main() { int a,b=3; a=square(b+2); printf("%d", a); }

- a. 25 b. 9 c. **11** d. Garbage

Q. 9 #define MAIN main() #define BEGIN { #define END } MAIN BEGIN printf("Good Morning"); END

- a. **You can not redefine main** b. No output c. Good Morning d. Compilation error

Q.10 What is Correct execution sequence?

- a. Compiler => Preprocessor => Linker b. Compiler => Linker => Preprocessor
c. **Preprocessor => Compiler => Linker** d. Preprocessor => Linker => Compiler

Q. 11 #define mult(x) x*x int mul(x) {return x*x; } void main() { int m,i=3; m=mul(i+2); printf("%d",m); m=mult(2+i); printf("%d",m); }

- a. 1125 b. **2511** c. 117 d. 257

Q.12#define P(format, var) printf("var=%format ",var); void main() { int i=3; float a=3.14; p(d,i); P(f,a); }

- a. var=3format var=3.140000ormat b. **var=0.000000ormat var=3.140000ormat**
c. var=3 var=3.140000ormat d. Error

Q. 13 #define D void main() { int i=2; #ifdef D printf("%d",i*=i); #else printf("%d",i); #endif }

- a. 2 b. **4** c. 16 d. Compilation error

Q.14 #define Ptr int * Void main() { Ptr P, Q; int a=3,b=5; P=&a; Q=&b; *P=*P + *Q; *Q=*P - *Q; *P=*P - *Q; printf("%d %d",a, b); }

C Programming Question Bank

a. 3 5 b. 5 3 c. 8 3 d. Error

Q. 15 #define is_____.

a. a pre-processor b. a macro itself c. a preprocessor directive d. a template

Q.16 #define prod(a,b) a*b void main() { int z, x=2,y=3; z=prod(x+2,y-1); printf("%d",z); }

a. 8 b. 7 c. 0 d. None of above

Q. 17

#define square(x, y) (x+y)

void main()

{

int i,a=4,b=3;

i=add(a+4,1+b);

printf("%d",i);

}

a. 14 b. 13 c. 12 d. Compilation error

Q.18 #define ROWS 3 #define COLS 4 int Z[ROW][COLS]={1,2,3,4,5,6,7,8,9,10,11,12};

void main() { int a,b,c=999; for(a=0;a

a. 12 b. 1 c. 999 d. None of these

Q.19

/* include file zzz.h starts here */ printf("Hello"); /* include file zzz.h ends here */ void main() { #include "zzz.h" printf("World"); }

a. World b. Hello c. Hello World d. Compilation error

Q.20 #define IMP #ifdef IMP int = 10; #endif; void main() { int = 5; printf("%d",IMP); }

a. 10 b. 0 c. 5 d. Error

Q. 21 #define fact(n) if(n>1) n*fact(n-1) void main() { int no=4; printf("%d",fact(no)); }

a. 4 b. 12 c. 24 d. Error

Q. 22 #define D void main() { int i=2; #ifdef D printf("%d",i*=i); #else printf("%d",i); #endif }

a. 2 b. 4 c. 16 d. Compilation error

Q.23 Macro definition can be extended on more than one lines using character_____.

a. / b. \ c. # d. %

Q.24

#define calc(a) (a*a*a); void main() { float i=2.0,p=0; p=calc(i); printf("%f", p); if(calc(i)= =8) printf("cube"); }

a. 8 b. 8.000000cube c. 8.000000 d. Compilation error

Q.25

#define NOW #define LATER void main() { #ifdef NOW & & LATER printf("We will go for movie"); #else printf("We will not go"); #endif }

a. We will not go b. We will go for movie c. Compilation error d. None

Q.26 Which of the following statement is true regarding macros and functions?

- a. Macros and functions are same
- b. Macros run program faster than functions**
- c. Due to functions the program size increases, hence macros are preferred
- d. Macros increases overheads of an activation record

Q. 27 The reason for discouraging macros in C is

- a. It slows down execution of executable code
- b. It expands the source code for compiler
- c. It makes code difficult to understand
- d. It increases memory requirements**

Q.28 #define together(s1,s2) s1##s2 void main() { int totsa1=599.89; printf("%d",together(tot,sal)); }

- a. 598.89
- b. 598**
- c. 599.0
- d. Compilation error

Q.29 #define result(a) (a+2*3) void main() { int x=3,y,z; y=result(x++); z=result(++x); printf("%d %d", y,z); }

- a. 9 11**
- b. 15 18
- c. 15 11
- d. 9 18

Q.30 Which of the following statement is Incorrect?

- a. Macros does not have address
- b. Macros does not perform data type checking
- c. Macros are handled by pre-processor
- d. Macros can not be redeclared in same file**

File Handling & Command Line Arguments

Q.1 Consider void main (int argc, char *argv[])

- a. argv[0] represents the command name that invokes the program**
- b. All of these
- c. argv[0] gives us source file name
- d. argv[0] can be NULL for any program

Q.2 What are Command line arguments?

- a. Input passed to program from command line**
- b. Arguments given on single line
- c. Information passed to Operating System Program
- d. Arguments passed to first function defined in program

Q.3

void main()

{

```
FILE *fp;  
fp=fopen("a.dat","r");  
fseek(fp, 0, 2);  
printf("%d",ftell(fp));  
fcloseall( );
```

}

- a. 12
- b. 11**
- c. 13
- d. 9

Q.4 What is returned by function ftell()?

- a. **Current position of File pointer (offset)**
- b. ASCII value of Character Present at current position
- c. File size
- d. Character Present at current position.

Q. 5 What is the use of fseek function?

- a. **Positioning a File pointer**
- b. Searching a number in file
- c. Copying data in file
- d. Reading a data at specified location

Q.6 MS-DOS represents EOF by ASCII values_____.

- a. 255
- b. 256
- c. 0
- d. **26**

Q.7 Consider the following code

```
void main( )  
{  
    FILE *fp=fopen("file1", "r");  
}
```

If file is existing & found, fp will point to

- a. First character of that file
- b. NULL
- c. name of that file
- d. **A structure that contains information about that file**

Q.8

```
void main( )  
{  
    FILE *fp;    char ch= 'O';  
    fp=fopen("A.dat", "w");  
    fputc(ch,fp);  
    fputs("let's study together",fp);  
}
```

What will be contents of file "A.dat" after executing above program?

- a. O
- b. **O let's study together**
- c. let's study together
- d. Error in Program

Q.9

```
void function(FILE *fp )  
{  
    fp=fopen("dummy.dat", "w");  
}  
void main( )  
{  
    FILE *fp  
    function(fp);  
    fprintf(fp,"%s", "Hi !");  
    fclose(fp);  
}
```

C Programming Question Bank

```
}
```

what will "dummy.dat" contain?

- a. **Nothing** b. Hi c. Garbage d. Runtime error

Q.10 EOF is defined in _____ header file

- a. stdlib.h **b. stdio.h** c. stdfile.h d. conio.h

Q.11 Which of the following file opening mode deletes contents of file, if it already exists?

- a. "wb" b. "wb+" **c. All of these** d. "w"

Q.12 What is returned by macro feof on end of file?

- a. 0 b. 1 c. any non zero number [Feedback](#)

Q.13

```
void main( )
```

```
{
```

```
    FILE *fp;
```

```
    fp=fopen("C:\My Documents\a.dat","w");
```

```
    if(!fp) exist(1);
```

```
    fclose(fp);
```

```
}
```

point out error in the above code

- a. Path provided in function fopen()
b. The function fclose() must have the filename as parameter
c. The above code will compile successfully
d. The file should be closed before exit()

Q.14 Text files end with

- a. **EOF** b. '\r' c. '/0' d. NULL

Q.15

```
void main( )
```

```
{
```

```
    FILE *fp;
```

```
    fp=fopen("a.dat","r");
```

```
    fseek(fp,-1, 2);
```

```
    printf("%c",fgetc(fp));
```

```
    fclose(fp);
```

```
}
```

- a. **i** b. EOF c. h d. garbage

Q.16

```
void main( )
```

```
{
```

```
    FILE *fp;
```

```
    char ch=0,cnt=0;
```

```
    fp=fopen("a.dat","r");
```

C Programming Question Bank

```
while((ch=fgetc(fp))!=EOF) cnt++;  
printf("%d bytes",cnt);  
}
```

- a. 10 bytes b. 13 bytes c. 9 bytes **d. 11 bytes**

Q. 17 Which are valid opening modes for binary files?

- a. rb **b. all of these** c. wb d. ab+

Q.18 What will happen if specified file does not exist while opening that file in program(fopen)?

a. Different Action will be taken Depending on file opening mode

- b. Function fopen will return NULL
c. It will create new File
d. It will generate Runtime Error

Q.19 What is NULL?

- a. It is end of file
b. It is a function defined in file "error.h"
c. It is a character

d. It is a macro defined in file "stdio.h"

Q.20 Which Function can handle Command line Arguments?

- a. that is defined as main** b. that is declared first in program
c. that is defined last in program d. that is defined first in program

Q.21 What are disk I/O functions?

a. Functions, which perform I/O operations on a Disk

- b. Functions, which perform I/O operations at various ports
c. Functions, which perform I/O operations to receive input from keyboard
d. Functions, which perform I/O operations to write output on VDU

Q.22 What will be size of file b.dat created in following program?

```
void main( )  
{  
    FILE *fp;  
    struct date  
{  
        int dd,mm,yy;  
    }  
  
    today={12345, 12345, 12345};  
    fp=fopen("b.dat","wb");  
    fwrite(&today,sizeof(today),1,fp);  
    fclose(fp);  
}
```

- a. 15 bytes b. 18 bytes **c. 6 bytes** d. 17 bytes

Q.23 What header files contain?

- a. Declaration of user defined functions **b. Prototypes of standard library functions**
c. Definition of user defined functions d. Definition of standard library functions

C Programming Question Bank

Q.24 Which of the following can not take file pointer to beginning Of File?

- a. fopen **b. fflush** c. fseek d. rewind

Q.25 Which one function can be used to close all files f1, f2, f3 having their file pointers fp1, fp2, fp3 respectively?

- a. fcloseall();** b. fclose(f1,f2,f3); c. fclose(); d. fclose(fp1,fp2,fp3);

Q.26 What will be size of file a.dat created in following program?

```
void main( )
{
    FILE *fp;
    struct date
    {
        int dd,mm,yy;
    }
    today={12345, 12345, 12345};
    fp=fopen("a.dat","w");
    fprintf(fp,"%d,%d,%d", today.dd, today.mm, today.yy);
    fclose(fp);
}
```

- a. 15 bytes b. 6 bytes c. 17 bytes **d. 18 bytes**

Q.27 Which of the following statement writes a character 'ch' to a printer?

- a. fputc(ch, stdprn);** b. putch(ch, stdprn); c. putch(ch, printer); d. fputc(ch, prn);

Q.28 What is true about argc?

- a. argv gives count of arguments b. argc & argv both gives count of arguments
c. argv[0] gives count of arguments **d. argc gives count of arguments**

Q. 29 What is not a parameter to fseek?

- a. filename** b. reference position c. FILE ptr d. offset

Q.30 Which function can be used to write entire structure in binary file?

- a. fputc b. fprintf c. fputs **d. fwrite**

Q.31

```
void main( )
{
    unsigned char ch;
    FILE *fp;
    fp=fopen("alpha.dat", "r");
    while((ch=fgetc(fp))!=EOF) putchar(ch);
    fclose(fp);
}
```

- a. It will display contents of file alpha.dat & then it will go in infinite loop
b. It will go in infinite loop
c. Error

C Programming Question Bank

d. It will display contents of file alpha.dat

Q. 32 Which Standard Function/Macro is used to check end of file?

- a. **feof** b. fend c. eof d. EOF

Q.33

```
void main( )
```

```
{
```

```
    char ch;
```

```
    chrscr( );
```

```
    ch=getche( );
```

```
    fprintf(stdout,"%c", ch);
```

```
}
```

While executing above code, 's' is typed then what will be seen on screen?

- a. No output **b. s s** c. Garbage d. s

Q.34 Which header file is required for using file handling functions?

- a. conio.h **b. stdio.h** c. stdfile.h d. stdlib.h

Q. 35 rewind (fp) function behaves same as

- a. **fseek(fp,0,0);** b. fseek(fp,0,1); c. fseek(fp,1,2); d. fseek(fp,0,2);

Q.36 Which is a wrong statement regarding file opening modes?

- a. In "r+" mode file may not exist b. In "w" mode file may not exist
c. In "a+" mode file may not exist d. In "r" mode file must exist

Q. 37 Which files are automatically associated with C program?

- a. standard input
b. standard input, standard output and standard error
c. standard output

Q.38 Which statement is true about argv?

- a. It is an array of character pointers b. It is an array of strings
c. All of these d. It is a pointer to an array of character pointers

Q.39 Consider a file "A.dat" containing text

"Today is going to be a great day!"

```
void main( )
```

```
{
```

```
    FILE *fp;  char str[80] ;
```

```
    fp=fopen("A.dat", "r");
```

```
    while(fgets(str,80,fp)!=NULL) puts(str);
```

```
}
```

- a. While loop is a infinite loop b. Only 'Today' is printed on screen
c. Error in Program **d. Contents of file are printed on screen**

Q.40. #include<stdio.h>

```
void main( )
```

```
{  
    printf("%#X",EOF);  
}
```

a. 0XFFFF

b. 0xFF

c. FF

d. 0X10

SMVITA