Bit-1

What is the output

#include<stdio.h>

int main(){

int x=123;

int i={

printf("c" "++")

};

for(x=0;x<=i;x++){

printf("%x ",x);

}

return 0;

}

Bit-2:

#include<stdio.h>

int i=40;

extern int i;

int main(){

do{

printf("%d",i++);

}

while(5,4,3,2,1,0);

return 0;

}

Bit-3:

#include<stdio.h>

int main()

{

int i=2,j=2;

while(i+1?--i:j++)

printf("i=%d j=%d",i,j);

return 0;

}

Bit-4:

void main()

{

int i=1;

for(;i++;)

printf("%d",i);

}

Bit-5:

#include<stdio.h>

int main()

{

int i,j;

i=j=2,3; // line 5

while(--i&&j++)

printf("%d %d\n",i,j);

printf("%d %d\n",i,j);

return 0;

}

Bit-6:

#include<stdio.h>

void main()

{

int i;

for(i=1,printf("Intialization");printf("\nCondition"),i++ <= 5;printf("%d",i))

printf("\nInside the loop\n");

}

Bit-7:

int z,x=5,y=-10,a=4,b=2;

z = x++ - --y \* b / a; What is the value of 'z'

Bit-8:

char\* myFunc (char \*ptr)

{

ptr += 3;

return (ptr);

}

int main()

{

char \*x, \*y;

x = "HELLO";

y = myFunc (x);

printf ("y = %s \n", y);

return 0;

}

Bit-9

int a=10,b;

b=a++ + ++a;

printf("%d,%d,%d,%d",b,a++,a,++a);

Bit-10:

void myFunc (int x)

{

if (x > 0)

myFunc(--x);

printf("%d, ", x);

}

int main()

{

myFunc(5);

return 0;

}

Bit-11:

int i,j;

int ctr = 0;

int myArray[2][3];

for (i=0; i<3; i++)

for (j=0; j<2; j++)

{

myArray[j][i] = ctr;

++ctr;

}

What is the value of myArray[1][2]; in the sample code above?

Bit 12:

char \*ptr;

char myString[] = "abcdefg";

ptr = myString;

ptr += 5;

What string does ptr point to in the sample code above?

Bit-13:

int x = 0;

for (x=1; x<4; x++);

printf("x=%d\n", x);

What will be printed when the sample code above is executed?

Bit -14:

int x = 5;

int y = 2;

char op = '\*';

switch (op)

{

default : x += 1;

case '+' : x += y; /\*It will go to all the cases\*/

case '-' : x -= y;

}

After the sample code above has been executed, what value will the variable x contain?

Bit-15:

short testarray[4][3] = { {1}, {2, 3}, {4, 5, 6} };

printf( "%d\n", sizeof( testarray ) );

Assuming a short is two bytes long, what will be printed by the above code?

Bit-16:

int y[4] = {6, 7, 8, 9};

int \*ptr = y + 2;

printf("%d\n", ptr[ 1 ] );

Bit-17:

char \*buffer = "0123456789";

char \*ptr = buffer;

ptr += 5;

printf( "%s\n", ptr );

printf( "%s\n", buffer );

Bit-18:

#include <stdio.h>

void func()

{

int x = 0;

static int y = 0;

x++; y++;

printf( "%d -- %d\n", x, y );

}

int main()

{

func();

func();

return 0;

}

Bit-19:

int m = -14;

int n = 6;

int o;

o = m % ++n;

n += m++ - o;

m <<= (o ^ n) & 3;

Bit 20:

short int x; /\* assume x is 16 bits in size \*/

What is the maximum number that can be printed using printf("%d\n", x), assuming that x is initialized as shown above?

Bit-21:

char txt [20] = "Hello world!\0";

How many bytes are allocated by the definition above?

Bit-22:

int x = 0;

for ( ; ; )

{

if (x++ == 4)

break;

continue;

}

printf("x=%d\n", x);

What will be printed when the sample code above is executed?

Bit-23:

#include

main(){

int num[] = {1,4,8,12,16};

int \*p,\*q;

int i;

p = num;

q = num+2;

i = \*p++;

printf("%d, %d, %d\n",i, \*p, \*q);

}

Bit 24:

main(){

char \*a[4]={"jaya","mahesh","chandra","swapant"};

int i = sizeof(a)/sizeof(char \*);

printf("i = %d\n", i);

}

Bit 25:

char s1[20];

char \*s = "fyc";

printf("%d %d\n", sizeof(s1), sizeof(s));

Bit 26:

int x=8,y=1;

switch (x--,y++)

{

case 1: x \*= 8;

case 2:y \*= x /= 2;

case 3:

case 4:y--;

default: x += 5;

}

printf("%d %d", x, y);

Bit 27:

What is the output of the following program?

#include<stdio.h>

void main()

{

int x=10,y=20;

x=((x>9) && y=30 ? printf("\nTRUE"): printf("\nFALSE"));

printf("\ny=%d",y);

}