Sample c questions and answers with explanation for interview

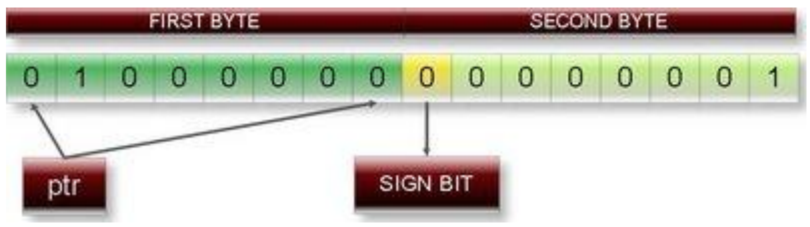
1.
What will be output if you will compile and execute the following c code?

```
void main(){
   int i=320;
   char *ptr=(char *)&i;
   printf("%d",*ptr);
}
```

(A) 320 ○
(B) 1 ○
(C) 64 ○
(D) Compiler error ○
(E) None of above ○

Explanation:

As we know size of int data type is two byte while char pointer can pointer one byte at time.
Memory representation of int i=320



So char pointer ptr is pointing to only first byte as shown above figure.

*ptr i.e. content of first byte is 01000000 and its decimal value is 64.
How to represent char, int and float data in memory?
Data type tutorial.

2.

What will be output if you will compile and execute the following c code?

```c
#define x 5+2
void main(){
    int i;
    i=x*x*x;
    printf("%d",i);
}
```

(A) 343 ○
(B) 27 ○
(C) 133 ○
(D) Compiler error ○
(E) None of above ○

Explanation:

As we know #define is token pasting preprocessor it only paste the value of micro constant in the program before the actual compilation start. If you will see intermediate file you will find:
test.c 1:

```
test.c 2: void main(){
test.c 3: int i;
test.c 4: i=5+2*5+2*5+2;
test.c 5: printf("%d",i);
test.c 6: }
test.c 7:
```

You can absorb #define only pastes the 5+2 in place of x in program. So,

```
i=5+2*5+2*5+2
=5+10+10+2
=27
```

What is intermediate file and how to see intermediate file?
Preprocessor tutorial.

3.

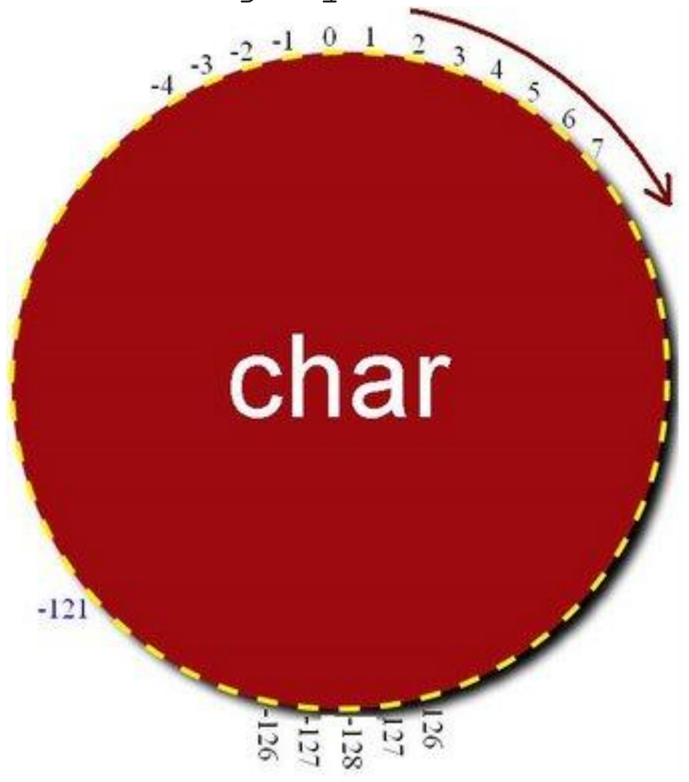
What will be output if you will compile and execute the following c code?

```
void main(){
char c=125;
    c=c+10;
    printf("%d",c);
}
```

(A) 135 ○
(B) +INF ○
(C) -121 ○
(D) -8 ○
(E) Compiler error ○

Explanation:

As we know char data type shows cyclic properties i.e. if you will increase or decrease the char variables beyond its maximum or minimum value respectively it will repeat same value according to following cyclic order:



So,
125+1= 126
125+2= 127
125+3=-128
125+4=-127
125+5=-126
125+6=-125
125+7=-124
125+8=-123

125+9=-122
125+10=-121
What is cyclic nature of data type?
Data type tutorial.

4.

What will be output if you will compile and execute the following c code?

```
void main(){
    float a=5.2;
  if(a==5.2)
     printf("Equal");
  else if(a<5.2)
     printf("Less than");
  else
     printf("Greater than");
}
```

(A) Equal ○
(B) Less than ○
(C) Greater than ○
(D) Compiler error ○
(E) None of above ○

Explanation:
5.2 is double constant in c. In c size of double data is 8 byte while a is float variable. Size of float variable is 4 byte.

So double constant 5.2 is stored in memory as:
101.00 11001100 11001100 11001100 11001100 11001100 11001101
Content of variable a will store in the memory as:
101.00110 01100110 01100110
It is clear variable a is less than double constant 5.2
Since 5.2 is recurring float number so it different for float and double. Number likes 4.5, 3.25, 5.0 will store same values in float and double data type.
Note: In memory float and double data is stored in completely different way. If you want to see actual memory representation goes to question number (60) and (61).
Data type tutorial.

5.

What will be output if you will compile and execute the following c code?

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

(A) 21 ○
(B) 18 ○

(C) 12 ○

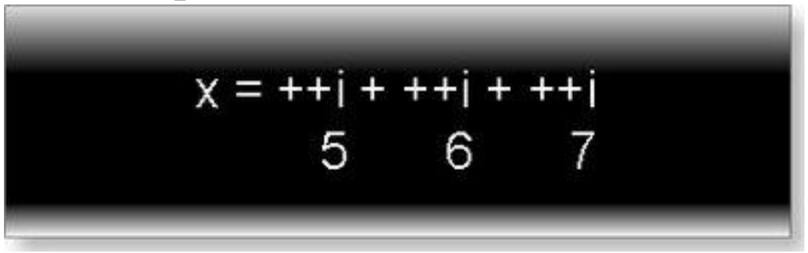
(D) Compiler error ○

(E) None of above ○

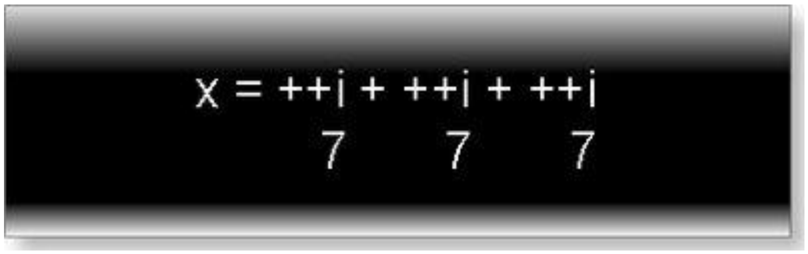
Explanation:

In ++a, ++ is pre increment operator. In any mathematical expression pre increment operator first increment the variable up to break point then starts assigning the final value to all variable.

Step 1: Increment the variable I up to break point.



Step 2: Start assigning final value 7 to all variable i in the expression.



So, i=7+7+7=21

What is break point?

Operator tutorial.

6.

What will be output if you will compile and execute the following c code?

```
void main(){
 int a=2;
 if(a==2){
   a=~a+2<<1;
   printf("%d",a);
 }
 else{
  break;
 }
}
```

(A) It will print nothing. ○
(B) -3 ○
(C) -2 ○
(D) 1 ○
(E) Compiler error ○

Explanation:

Keyword break is not part of if-else statement. Hence it will show compiler error: Misplaced break
Where we can use break keyword?
Control statement tutorial

7.

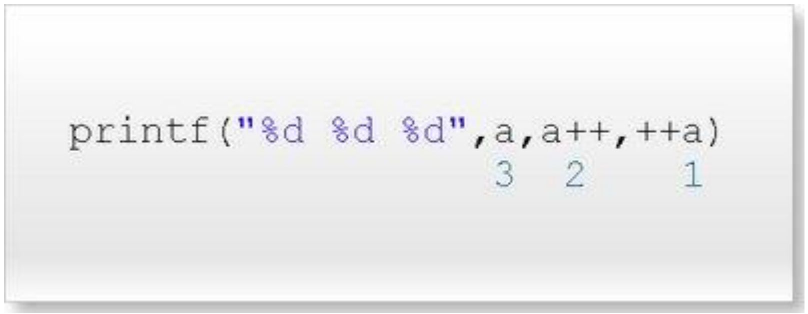
What will be output if you will compile and execute the following c code?

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

(A) 12 11 11 ○
(B) 12 10 10 ○
(C) 11 11 12 ○
(D) 10 10 12 ○
(E) Compiler error ○

Explanation:

In c printf function follows cdecl parameter passing scheme. In this scheme parameter is passed from right to left direction.



```
printf("%d %d %d",a,a++,++a)
                  3  2    1
```

So first ++a will pass and value of variable will be a=10 then a++ will pass now value variable will be a=10 and at the end a will pass and value of a will be a=12.

What is cedecl and pascal parameter passing convention?

Function tutorial.

8.

What will be output if you will compile and execute the following c code?

```
void main(){
   char *str="Hello world";
   printf("%d",printf("%s",str));
}
```

(A) 11Hello world ○
(B) 10Hello world ○
(C) Hello world10 ○
(D) Hello world11 ○
(E) Compiler error ○

Explanation:
Return type of printf function is integer and value of this integer is exactly equal to number of character including white space printf function prints. So, printf("Hello world") will return 13.
What is prototype of printf function?
Formatted I/O tutorial.

9.

What will be output if you will compile and execute the following c code?

```
#include "stdio.h"
```

```
#include "string.h"
void main(){
   char *str=NULL;
   strcpy(str,"cquestionbank");
   printf("%s",str);
}
```

(A) cquestionbank ○
(B) cquestionbank\0 ○
(C) (null) ○
(D) It will print nothing ○
(E) Compiler error ○

Explanation:

We cannot copy any thing using strcpy function to the character pointer pointing to NULL.

String tutorial.

More questions of string.

10.

What will be output if you will compile and execute the following c code?

```
#include "stdio.h"
#include "string.h"
void main(){
  int i=0;
  for(;i<=2;)
   printf(" %d",++i);
}
```

(A) 0 1 2 ○

(B) 0 1 2 3 ○

(C) 1 2 3 ○

(D) Compiler error ○

(E) Infinite loop ○

Explanation:

In for loop each part is optional.
Complete tutorial of looping in C.

11.

What will be output if you will compile and execute the following c code?

```
void main(){
  int x;
  for(x=1;x<=5;x++);
    printf("%d",x);
}
```

(A) 4 ○

(B) 5 ○

(C) 6 ○

(D) Compiler error ○

(E) None of above ○

Explanation:

Body of for loop is optional. In this question for loop will execute until value of variable x became six and condition became false.

Looping tutorial.

12.

What will be output if you will compile and execute the following c code?

```
void main(){
printf("%d",sizeof(5.2));
}
```

(A) 2 ○
(B) 4 ○
(C) 8 ○
(D) 10 ○
(E) Compiler error ○

Explanation:

Default type of floating point constant is double. So 5.2 is double constant and its size is 8 byte.

Detail explanation of all types of constant in C.

13.

What will be output if you will compile and execute the following c code?

```
#include "stdio.h"
#include "string.h"
void main(){
char c='\08';
printf("%d",c);
}
```

(A) 8 ○
(B) '8' ○
(C) 9 ○
(D) null ○
(E) Compiler error ○

Explanation:

In c any character is starting with character '\' represents octal number in character. As we know octal digits are: 0, 1, 2, 3, 4, 5, 6, and 7. So 8 is not an octal digit. Hence '\08' is invalid octal character constant.

Octal character constantan.
Hexadecimal character constant.

14.

What will be output if you will compile and execute the following c code?

```
#define call(x,y) x##y
void main(){
int x=5,y=10,xy=20;
printf("%d",xy+call(x,y));
}
```

(A) 35 ○
(B) 510 ○
(C) 15 ○
(D) 40 ○
(E) None of above ○

Explanation:

## is concatenation c preprocessor operator. It only concatenates the operands i.e.
a##b=ab
If you will see intermediate file then you will find code has converted into following intermediate code before the start of actual compilation.
Intermediate file:

```
test.c 1:
test.c 2: void main(){
test.c 3: int x=5,y=10,xy=20;
test.c 4: printf("%d",xy+xy);
test.c 5: }
```

test.c 6:
It is clear call(x, y) has replaced by xy.
What is macro call?
Preprocessor tutorial.

15.

What will be output if you will compile and execute the following c code?

```
int * call();
void main(){
int *ptr;
ptr=call();
    clrscr();
printf("%d",*ptr);
}
int * call(){
int a=25;
a++;
return &a;
}
```

(A) 25 ○
(B) 26 ○
(C) Any address ○
(D) Garbage value ○
(E) Compiler error ○

Explanation:

In this question variable a is a local variable and its scope and visibility is within the function call. After returning the address of a by function call variable a became dead while pointer ptr is still pointing to address of variable a. This problem is known as dangling pointer problem.
Complete pointer tutorial.

16.

What is error in following declaration?

```
struct outer{
int a;
struct inner{
char c;
};
};
```

(A) Nesting of structure is not allowed in c. ○

(B) It is necessary to initialize the member variable. ○

(C) Inner structure must have name. ○

(D) Outer structure must have name. ○

(E) There is not any error. ○

Explanation:

It is necessary to assign name of inner structure at the time of declaration other wise we cannot access the member of inner structure. So correct declaration is:

```
struct outer{
int a;
struct inner{
char c;
}name;
};
```

Structure tutorial.
Union tutorial.

17.

What will be output if you will compile and execute the following c code?

```
void main(){
int array[]={10,20,30,40};
printf("%d",-2[array]);
}
```

(A) -60 ○
(B) -30 ○
(C) 60 ○
(D) Garbage value ○
(E) Compiler error ○

Explanation:
In c,

array[2]=*(array+2)=*(2+array)=2[array]=30
Array tutorial.
Array of pointer.
How to read complex pointers.

18.

What will be output if you will compile and execute the following c code?

```
void main(){
int i=10;
static int x=i;
if(x==i)
printf("Equal");
else if(x>i)
printf("Greater than");
else
printf("Less than");
}
```

(A) Equal ○
(B) Greater than ○
(C) Less than ○
(D) Compiler error ○
(E) None of above ○

Explanation:
static variables are load time entity while auto variables are run time entity. We can

not initialize any load time variable by the run time variable.
In this example i is run time variable while x is load time variable.
What is storage class?

19.

What will be output if you will compile and execute the following c code?

```
#define max 5;
void main(){
int i=0;
i=max++;
printf("%d",i++);
}
```

(A) 5 ○
(B) 6 ○
(C) 7 ○
(D) 0 ○
(E) Compiler error ○

Explanation:

#define is token pasting preprocessor. If you will see intermediate file: test.i

```
test.c 1:
test.c 2: void main(){
test.c 3: int i=0;
test.c 4: i=5++;
test.c 5: printf("%d",i++);
test.c 6: }
test.c 7:
```

It is clear macro constant max has replaced by 5. It is illegal to increment the constant number. Hence compiler will show Lvalue required.

What is Lvalue and Rvalue?
How to see intermediate file?
Preprocessor questions and answer.

20.

What will be output if you will compile and execute the following c code?

```
void main(){
double far* p,q;
printf("%d",sizeof(p)+sizeof q);
}
```

(A) 12 ○
(B) 8 ○
(C) 4 ○
(D) 1 ○
(E) Compiler error ○

Explanation:

It is clear p is far pointer and size of far pointer is 4 byte while q is double variable and size of double variable is 8 byte.

What is near pointer?
Complete poin
What is far pointer?
What is huge pointer?ter tutorial.

If you have any queries in above c questions answers you can ask here.

Tricky c programs question for interview and answers with explanation. These questions are for experienced persons.

C advanced interview questions and answers

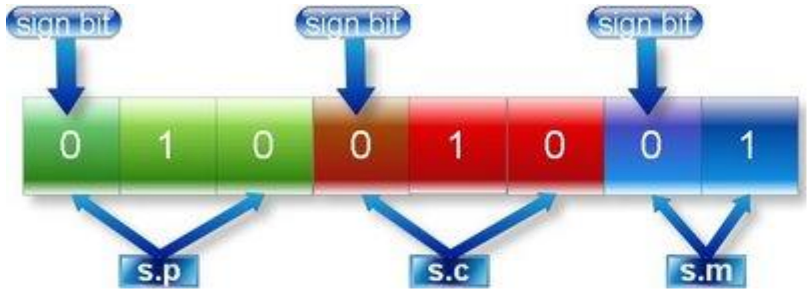
(1) What will be output if you will compile and execute the following c code?

```
struct marks{
  int p:3;
  int c:3;
  int m:2;
};
void main(){
  struct marks s={2,-6,5};
  printf("%d %d %d",s.p,s.c,s.m);
}
```

(a) 2 -6 5
(b) 2 -6 1
(c) 2 2 1
(d) Compiler error
(e) None of these

Answer: (c)
Explanation:
Binary value of 2: 00000010 (Select three two bit)
Binary value of 6: 00000110
Binary value of -6: 11111001+1=11111010
(Select last three bit)
Binary value of 5: 00000101 (Select last two bit)

Complete memory representation:



(2) What will be output if you will compile and execute the following c code?

```
void main(){
   int huge*p=(int huge*)0XC0563331;
   int huge*q=(int huge*)0xC2551341;
   *p=200;
   printf("%d",*q);
}
```

(a)0
(b)Garbage value
(c)null
(d) 200
(e)Compiler error

Answer: (d)
Explanation:
Physical address of huge pointer p
Huge address: 0XC0563331
Offset address: 0x3331
Segment address: 0XC056
Physical address= Segment address * 0X10 + Offset address
=0XC056 * 0X10 +0X3331
=0XC0560 + 0X3331
=0XC3891
Physical address of huge pointer q
Huge address: 0XC2551341
Offset address: 0x1341
Segment address: 0XC255
Physical address= Segment address * 0X10 + Offset address
=0XC255 * 0X10 +0X1341
=0XC2550 + 0X1341
=0XC3891
Since both huge pointers p and q are pointing same physical address so content of q will also same as content of q.

(3) Write c program which display mouse pointer and position of pointer.(In x coordinate, y coordinate)?

Answer:

```
#include"dos.h"
#include"stdio.h"
void main()
{
    union REGS i,o;
    int x,y,k;
    //show mouse pointer
    i.x.ax=1;
    int86(0x33,&i,&o);
while(!kbhit()) //its value will false when
we hit key in the key board
    {
    i.x.ax=3; //get mouse position
    x=o.x.cx;
    y=o.x.dx;
    clrscr();
    printf("(%d , %d)",x,y);
    delay(250);
    int86(0x33,&i,&o);
    }
    getch();
}
```

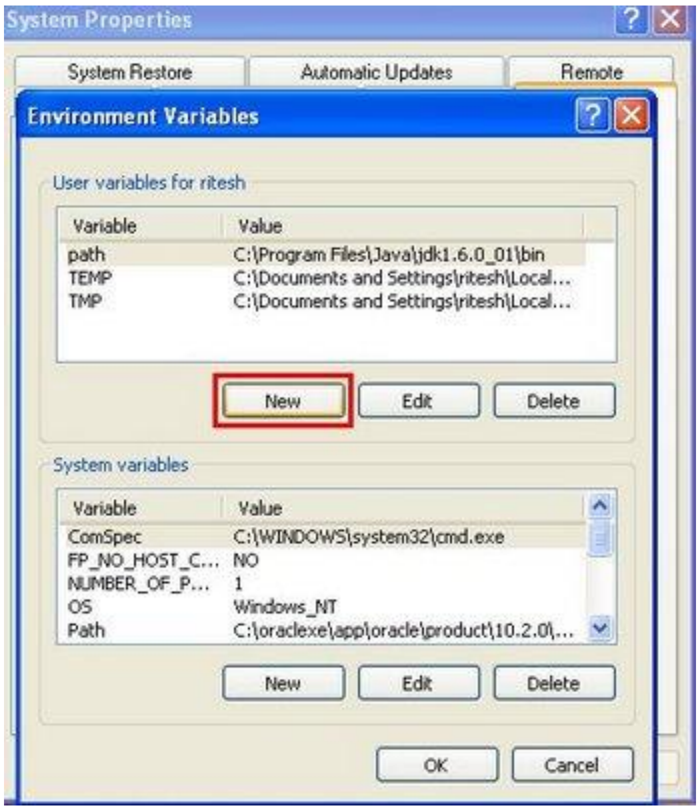
(4) Write a c program to create dos command: dir.

Answer:

Step 1: Write following code.

```
#include “stdio.h”
#include “dos.h”
void main(int count,char *argv[]){
   struct find_t q ;
   int a;
   if(count==1)
      argv[1]="*.*";
      a = _dos_findfirst(argv[1],1,&q);
      if(a==0){
         while (!a){
            printf(" %s\n", q.name);
            a = _dos_findnext(&q);
         }
      }
      else{
         printf("File not found");
      }
}
```

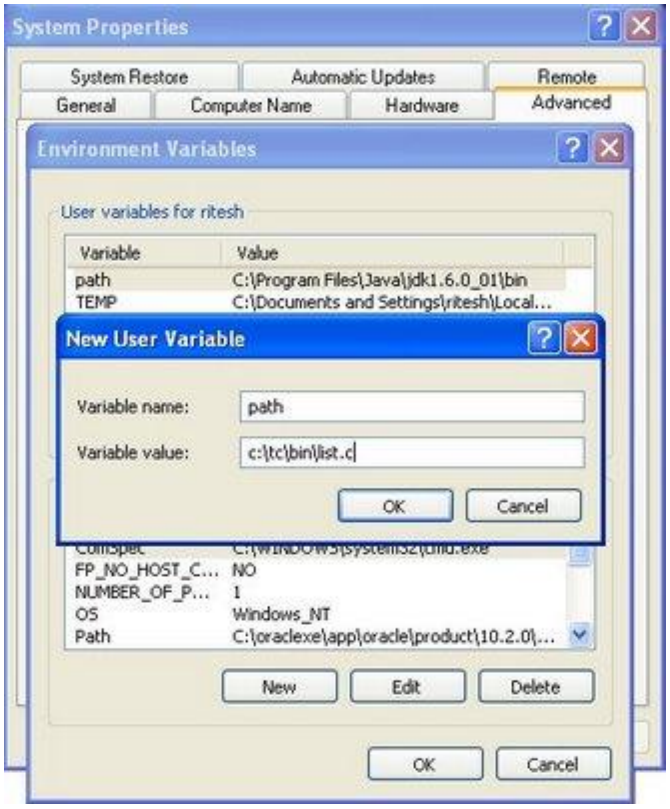
Step 2: Save the as list.c (You can give any name)
Step 3: Compile and execute the file.
Step 4: Write click on My computer of Window XP operating system and select properties.
Step 5: Select Advanced -> Environment Variables
Step 6: You will find following window:
Click on new button (Button inside the red box)



Variable	Value
path	C:\Program Files\Java\jdk1.6.0_01\bin
TEMP	C:\Documents and Settings\ritesh\Local...
TMP	C:\Documents and Settings\ritesh\Local...

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\oraclexe\app\oracle\product\10.2.0\...

Step 7: Write following:
Variable name: path
Variable value: c:\tc\bin\list.c (Path where you have saved)



Variable	Value
path	C:\Program Files\Java\jdk1.6.0_01\bin
TEMP	C:\Documents and Settings\ritesh\Local...

Variable name: path

Variable value: c:\tc\bin\list.c

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	1
OS	Windows_NT
Path	C:\oraclexe\app\oracle\product\10.2.0\...

Step 8: Open command prompt and write list and press enter.
Command line argument tutorial.

(6) What will be output if you will compile and execute the following c code?

```
void main(){
    int i=10;
    static int x=i;
    if(x==i)
        printf("Equal");
    else if(x>i)
        printf("Greater than");
    else
        printf("Less than");
}
```

(a) Equal
(b) Greater than
(c) Less than
(d) Compiler error
(e) None of above

Answer: (d)
Explanation:
static variables are load time entity while auto variables are run time entity. We can not initialize any load time variable by the run time variable.
In this example i is run time variable while x is load time variable.
Properties of static variables.
Properties of auto variables.

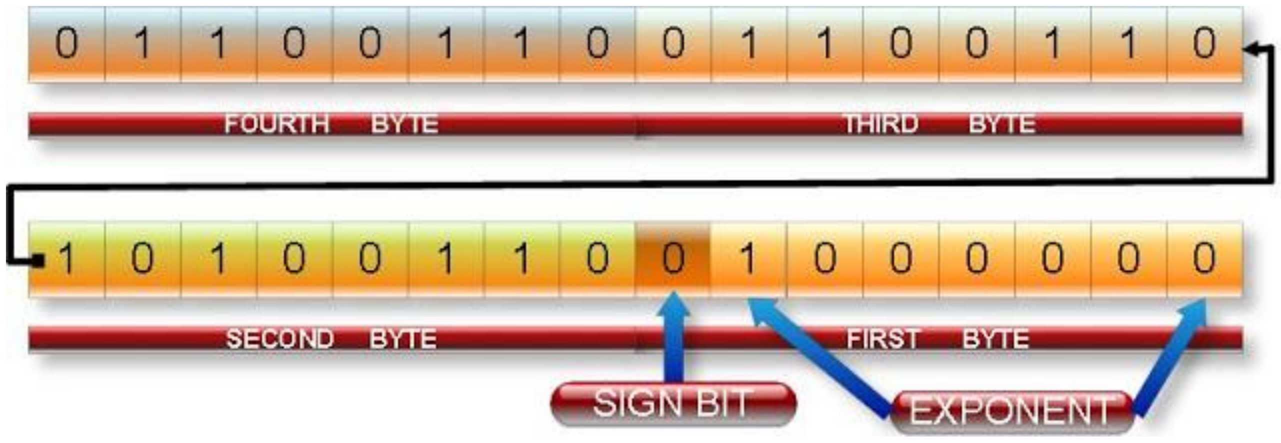
(7) What will be output if you will compile and execute the following c code?

```
void main(){
   int i;
   float a=5.2;
   char *ptr;
   ptr=(char *)&a;
   for(i=0;i<=3;i++)
      printf("%d ",*ptr++);
}
```

(a)0 0 0 0
(b)Garbage Garbage Garbage Garbage
(c)102 56 -80 32

(d)102 102 -90 64
(e)Compiler error

Answer: (d)
Explanation:
In c float data type is four byte data type while char pointer ptr can point one byte of memory at a time.
Memory representation of float a=5.2



ptr pointer will point first fourth byte then third byte then second byte then first byte.
Content of fourth byte:
Binary value=01100110
Decimal value= 64+32+4+2=102
Content of third byte:
Binary value=01100110
Decimal value=64+32+4+2=102
Content of second byte:
Binary value=10100110
Decimal value=-128+32+4+2=-90
Content of first byte:

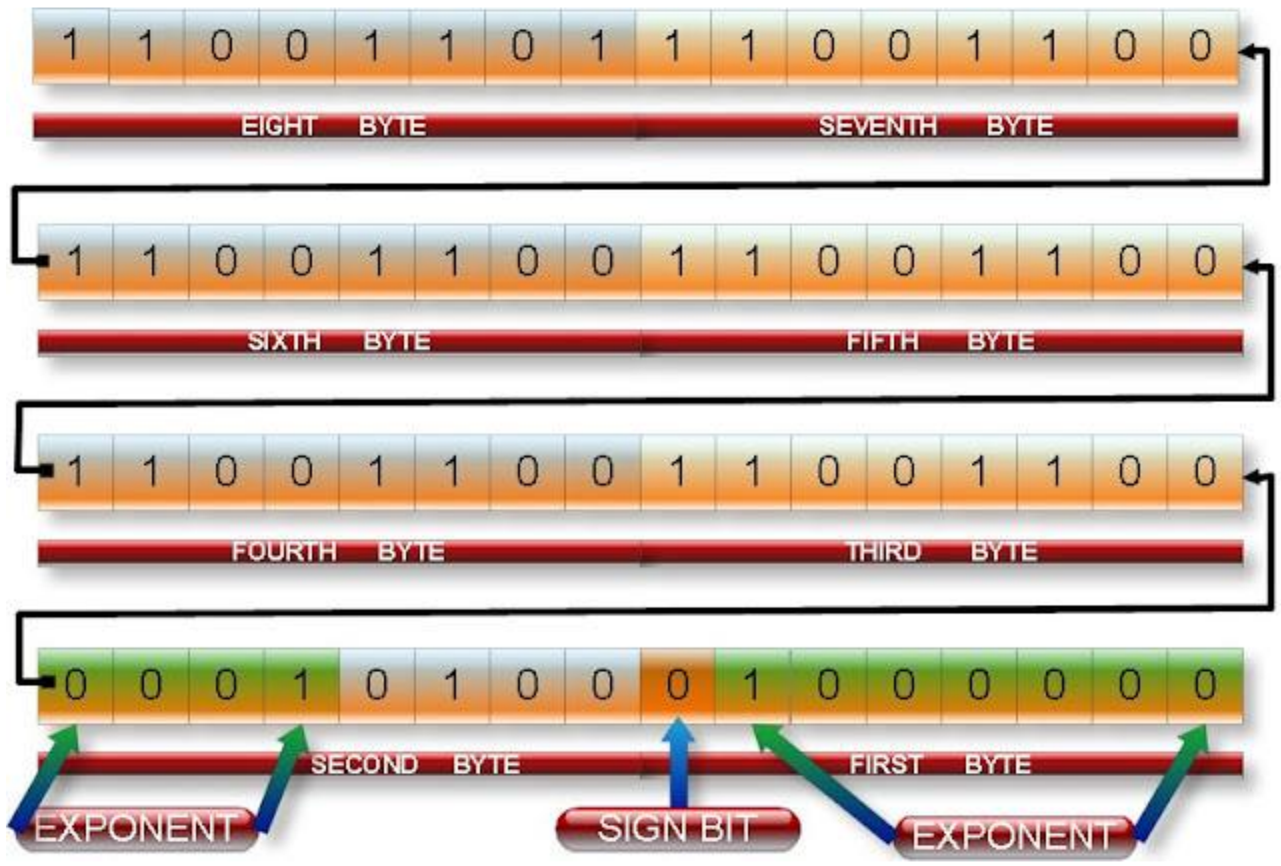
Binary value=01000000
Decimal value=64
Note: Character pointer treats MSB bit of each byte i.e. left most bit of above figure as sign bit.

(8) What will be output if you will compile and execute the following c code?

```c
void main(){
   int i;
   double a=5.2;
   char *ptr;
   ptr=(char *)&a;
   for(i=0;i<=7;i++)
      printf("%d ",*ptr++);
}
```

(a) -51 -52 -52 -52 -52 -52 20 64
(b) 51 52 52 52 52 52 20 64
(c) Eight garbage values.
(d) Compiler error
(e) None of these

Answer: (a)
Explanation:
In c double data type is eight byte data type while char pointer ptr can point one byte of memory at a time.
Memory representation of double a=5.2



ptr pointer will point first eighth byte then seventh byte then sixth byte then fifth byte then fourth byte then third byte then second byte then first byte as shown in above figure.

Content of eighth byte:

Binary value=11001101

Decimal value= -128+64+8+4+1=-51

Content of seventh byte:

Binary value=11001100

Decimal value= -128+64+8+4=-52

Content of sixth byte:

Binary value=11001100

Decimal value= -128+64+8+4=-52

Content of fifth byte:
Binary value=11001100
Decimal value= -128+64+8+4=-52
Content of fourth byte:
Binary value=11001100
Decimal value= -128+64+8+4=-52
Content of third byte:
Binary value=11001100
Decimal value= -128+64+8+4=-52
Content of second byte:
Binary value=00010100
Decimal value=16+4=20
Content of first byte:
Binary value=01000000
Decimal value=64
Note: Character pointer treats MSB bit of each byte i.e. left most bit of above figure as sign bit.

(9) What will be output if you will compile and execute the following c code?

```
void main(){
   printf("%s","c" "question" "bank");
}
```

(a) c question bank
(b) c
(c) bank
(d) cquestionbank
(e) Compiler error

Answer: (d)

Explanation:
In c string constant “xy” is same as “x” “y”
String tutorial.

(10) What will be output if you will compile and execute the following c code?

```
void main(){
   printf("%s",__DATE__);
}
```

(a) Current system date
(b) Current system date with time
(c) null
(d) Compiler error
(e) None of these

Answer: (a)
Explanation:
__DATE__ is global identifier which returns current system date.

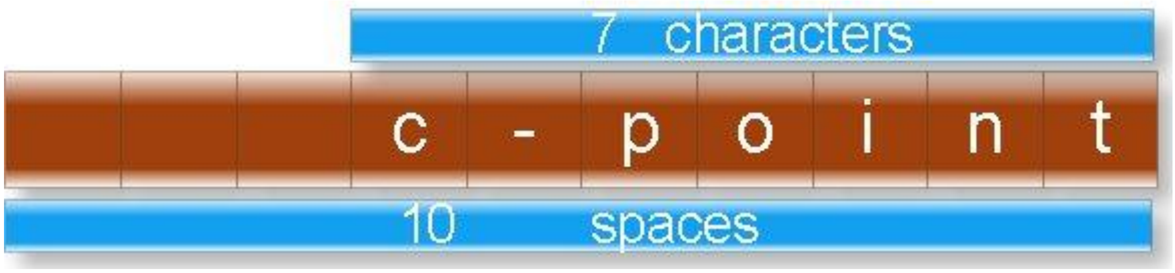
(11) What will be output if you will compile and execute the following c code?

```
void main(){
   char *str="c-pointer";
   printf("%*.*s",10,7,str);
}
```

(a) c-pointer
(b) c-pointer

(c) c-point
(d) cpointer null null
(e) c-point

Answer: (e)
Explanation:
Meaning of %*.*s in the printf function:
First * indicates the width i.e. how many spaces will take to print the string and second * indicates how many characters will print of any string.
Following figure illustrates output of above code:



Properties of printf function.

(12) What will be output if you will compile and execute the following c code?

```
void start();
void end();
#pragma startup start
#pragma exit end
int static i;
void main(){
   printf("\nmain function: %d",++i);
}
```

```
void start(){
   clrscr();
   printf("\nstart function: %d",++i);
}
void end(){
   printf("\nend function: %d",++i);
   getch();
}
```

(a)
main function: 2
start function: 1
end function:3
(b)
start function: 1
main function: 2
end function:3
(c)
main function: 2
end function:3
start function: 1
(d) Compiler error
(e) None of these

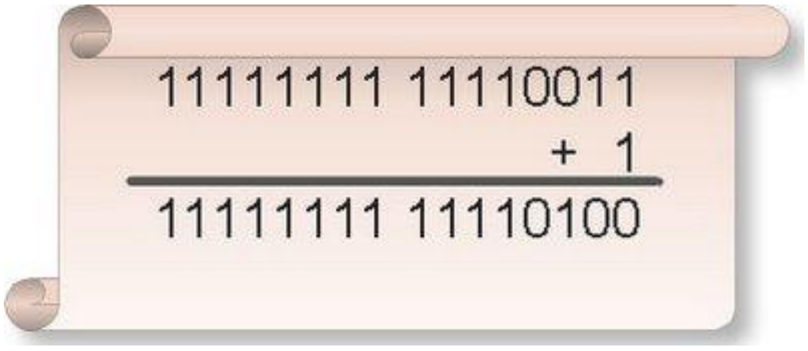
Answer: (b)
Explanation:
Every c program start with main function and terminate with null statement. But #pragma startup can call function just before main function and #pragma exit
What is pragma directive?
Preprocessor tutorial.

(13) What will be output if you will compile and execute the following c code?

```c
void main(){
   int a=-12;
   a=a>>3;
   printf("%d",a);
}
```

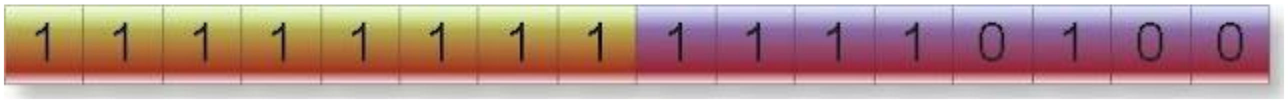
(a) -4
(b) -3
(c) -2
(d) -96
(e) Compiler error

Answer :(c)
Explanation:
Binary value of 12 is: 00000000 00001100
Binary value of -12 wills 2's complement of 12 i.e.



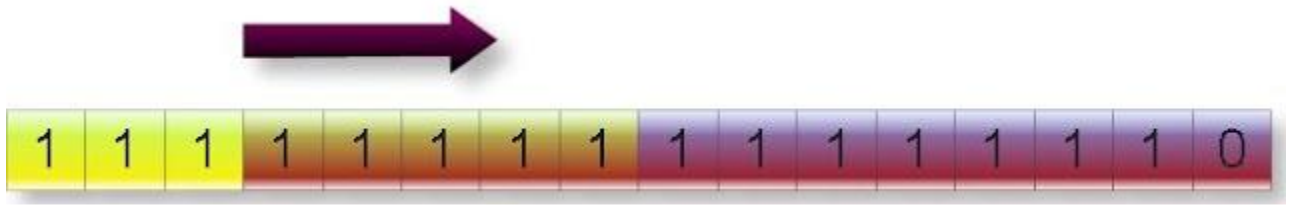
```
11111111 11110011
              + 1
-----------------
11111111 11110100
```

So binary value of -12 is: 11111111 11110100

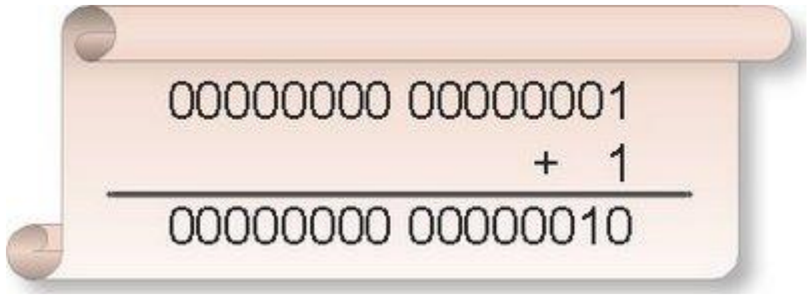


1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0

Right shifting rule:
Rule 1: If number is positive the fill vacant spaces in the left side by 0.
Rule 2: If number is negative the fill vacant spaces in the left side by 1.
In this case number is negative. So right shift all the binary digits by three space and fill vacant space by 1 as shown following figure:

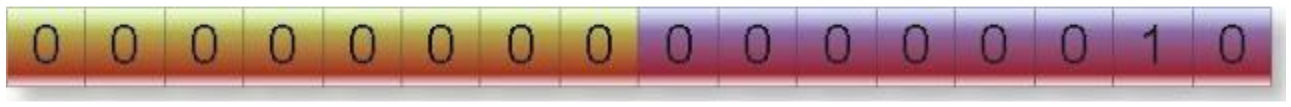


Since it is negative number so output will also a negative number but its 2’s complement.



```
00000000 00000001
              + 1
00000000 00000010
```

Hence final out put will be:



And its decimal value is: 2
Hence output will be:-2

(14) What will be output if you will compile and execute the following c code?

```
#include "string.h"
void main(){
   clrscr();
 printf("%d%d",sizeof("string"),strlen("str
ing"));
getch();
}
```

(a) 6 6
(b) 7 7
(c) 6 7
(d) 7 6
(e) None of these

Answer: (d)
Explanation:
Sizeof operator returns the size of string including null character while strlen function returns length of a string excluding null character.

(15) What will be output if you will compile and execute the following c code?

```
void main(){
   static main;
   int x;
   x=call(main);
   clrscr();
   printf("%d ",x);
   getch();
}
int call(int address){
   address++;
   return address;
}
```

(a) 0
(b) 1
(c) Garbage value
(d) Compiler error
(e) None of these

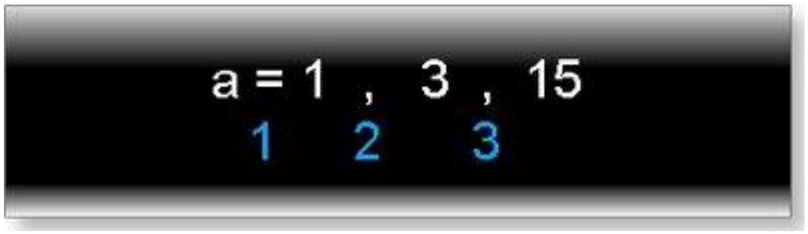
Answer: (b)
Explanation:
As we know main is not keyword of c but is special type of function. Word main can be name variable in the main and other functions.
What is main function in c?

(16) What will be output if you will compile and execute the following c code?

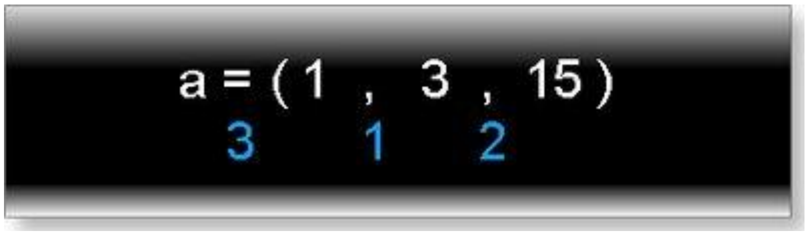
```
void main(){
   int a,b;
   a=1,3,15;
   b=(2,4,6);
   clrscr();
   printf("%d ",a+b);
   getch();
```

}
(a) 3
(b) 21
(c) 17
(d) 7
(e) Compiler error

Answer: (d)
Explanation:
In c comma behaves as separator as well as operator.
a=1, 3, 15;
b= (2, 4, 6);
In the above two statements comma is working as operator. Comma enjoys least precedence and associative is left to right.
Assigning the priority of each operator in the first statement:



Hence 1 will assign to a.
Assigning the priority of each operator in the second statement:



(17) What will be output if you will compile and execute the following c code?

```
int extern x;
void main()
   printf("%d",x);
   x=2;
   getch();
}
int x=23;
```

(a) 0
(b) 2
(c) 23
(d) Compiler error
(e) None of these

Answer: (c)
Explanation:
extern variables can search the declaration of variable any where in the program.

(18) What will be output if you will compile and execute the following c code?

```
void main(){
    int i=0;
    if(i==0){
      i=((5,(i=3)),i=1);
      printf("%d",i);
   }
   else
      printf("equal");
}
```

(a) 5
(b) 3
(c) 1
(d) equal
(e) None of above

Answer: (c)
Explanation:

(19) What will be output if you will compile and execute the following c code?

```
void main(){
   int a=25;
   clrscr();
   printf("%o %x",a,a);
   getch();
}
```

(a) 25 25
(b) 025 0x25
(c) 12 42
(d) 31 19
(e) None of these

Answer: (d)
Explanation:

%o is used to print the number in octal number format.
%x is used to print the number in hexadecimal number format.
Note: In c octal number starts with 0 and hexadecimal number starts with 0x.

(20) What will be output if you will compile and execute the following c code?

```
#define message "union is\
power of c"
void main(){
   clrscr();
   printf("%s",message);
   getch();
}
```

(a) union is power of c
(b) union ispower of c
(c) union is
Power of c
(d) Compiler error
(e) None of these

Answer: (b)
Explanation:
If you want to write macro constant in new line the end with the character \.

(21) What will be output if you will compile and execute the following c code?

```
#define call(x) #x
```

```
void main(){
   printf("%s",call(c/c++));
}
```

(a) c
(b) c++
(c) #c/c++
(d) c/c++
(e) Compiler error

Answer: (d)
Explanation:
# is string operator. It converts the macro function call argument in the string. First see the intermediate file:

```
test.c 1:
test.c 2: void main(){
test.c 3: printf("%s","c/c++");
test.c 4: }
test.c 5:
```

It is clear macro call is replaced by its argument in the string format.

(22) What will be output if you will compile and execute the following c code?

```
void main(){
   if(printf("cquestionbank"))
      printf("I know c");
   else
      printf("I know c++");
}
```

(a) I know c
(b) I know c++
(c) cquestionbankI know c
(d) cquestionbankI know c++
(e) Compiler error

Answer: (c)
Explanation:
Return type of printf function is integer which returns number of character it prints including blank spaces. So printf function inside if condition will return 13. In if condition any non- zero number means true so else part will not execute.

If you have any doubt in above **Tricky questions with explanation** you can ask through comment section.

C Linux interview questions and answers

(1)What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    int a=5;
    printf("%d %d %d",a++,a++,++a);
    return 0;
}
```

Output:
In LINUX GCC compiler

```
7 6 8
In TURBO C
7 6 6
```

Hints: In Turbo c parameter is passed from right to left in printf function but not in the Linux.

(2)What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    int a=5,b=10,c=15,d=20;
    printf("%d %d %d");
        return 0;
}
```

Output:
In LINUX GCC compiler
Garbage values
In TURBO C
5 10 15

Hints: Local variables stores in the stack.

(3) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    int i=5,j=5,y;
    int x=++i + ++i + ++i;
    y=++j + ++j + ++j;
    printf("%d  %d  %d %d",x,y,i,j);
    return 0;
}
```

Output:
In LINUX GCC compiler

```
22  22  8  8
In TURBO C
```

```
21  24  8  8
```

(4) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
        int  near *p;
        int  far *q;
        int  huge *r;
        printf("%d  %d
%d",sizeof(p),sizeof(q),sizeof(r));
        return 0;
}
```

Output:
In LINUX GCC compiler
Compilation error
In TURBO C

```
2  4  4
```

Note: In Linux there is not any concept of near, far and huge pointers

(5) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    char *p;
    int *q;
    float **r;
    printf("%d  %d  %d",sizeof(p),sizeof(q),sizeof(r));
    return 0;
}
```

Output:
In LINUX GCC compiler

```
4  4  4
```

In TURBO C

```
2  2  2
```

Hints: size of any type of pointer in Linux is 4 and in turbo c is 2.

(6) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    short int a=5;
    int b=5;
    long int c=5l;
    float d=5.0f;
    double e=5.0;
    long double f=5.0L;
    char g='5';
    printf("Size of short int: %d\n",sizeof(a));
    printf("Size of int: %d\n",sizeof(b));
    printf("Size of long int: %d\n",sizeof(c));
    printf("Size of float: %d\n",sizeof(d));
    printf("Size of double: %d\n",sizeof(e));
    printf("Size of long double: %d\n",sizeof(f));
    printf("Size of char: %d\n",sizeof(g));
        return 0;
}
```

Output:

In LINUX GCC compiler

```
Size of short int: 2
Size of int: 4
Size of long int: 4
Size of float: 4
Size of double: 8
Size of long double: 12
Size of char: 1
```

In TURBO C

```
Size of short int: 2
Size of int: 2
Size of long int: 4
Size of float: 4
Size of double: 8
Size of long double: 10
Size of char: 1
```

(7) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
    int main(){
        int a=300;
    char *p=(char *)&a;
        printf("%d\n",*p);
    printf("%d",*++p);
        return 0;
}
```

Output:
In LINUX GCC compiler

```
44
1
```

In TURBO C

```
44
1
```

(8) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    char c='A';
    printf("%d   %d",sizeof(c),sizeof('A'));
    return 0;
}
```

Output:
In LINUX

```
1   4
```

In TURBO C

```
1   2
```

(9) What will be output if you will execute following program by gcc compiler in Linux?

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

```
int main(){
    enum color{RED,BLUE,GREEN=-2,YELLOW,PINK};
    printf("%d  %d",BLUE,PINK);
    return 0;
}
```

Output:
In LINUX GCC compiler

```
1 0
```

In TURBO C

```
1 0
```

(10) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
    char c=127;
    printf("%d",++c);
    printf("  %d",++c);
    return 0;
}
```

Output:
In LINUX GCC compiler

```
-128 -127
```

In TURBO C

```
-128 -127
```

Hints: char data type cyclic property.

(11) What will be output if you will execute following program by gcc compiler in Linux?

```
#include"stdio.h"
struct info1{
    char *title;
    long int size;
    double grade;
}hero1;
union info2{
        char *title;
        long int size;
        double grade;
}hero2;
int main(){
    printf("Size of structure: %d\n",sizeof(hero1));
    printf("Size of union:  %d",sizeof(hero2));
    return 0;
}
```

Output:
In LINUX GCC compiler

```
Size of structure: 16
Size of union:  8
In TURBO C
Size of structure: 14
Size of union:  8
```

(12) What will be output if you will execute following program by gcc compiler in Linux?

```
#define size(x) (char *)(x+1)-(char *)x
#include<stdio.h>
int main(){
    long int *p;
    long double *q;
        printf("Size of long int: %d\n",size(p));
    printf("Size of long double: %d",size(q));
        return 0;
}
```

Output:
In LINUX GCC compiler

```
Size of long int: 4
Size of long double: 12
In TURBO C
Size of long int: 4
Size of long double: 10
```

(13) What will be output if you will execute following program by gcc compiler in Linux?

```
#include<stdio.h>
int main(){
      int i=2,j=5,k=3;
      int a=i&&j>=k;
      printf("%d",a);
      return 0;
}
```

Output:
In LINUX GCC compiler
1
In TURBO C
1
Hints: Any conditional or relational operator returns 1 if condition is true otherwise it returns 0.

If you have any queries or suggestions in above **c Linux interview questions**, please share it.

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<5;i++){
         int i=10;
         printf(" %d",i);
         i++;
    }
    return 0;
}
```

(A) 10 11 12 13 14 ○

(B) 10 10 10 10 10 ○

(C) 0 1 2 3 4 ○

(D) Compilation error ○

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    register a,b,x;
    scanf("%d %d",&a,&b);
    x=a+~b;
    printf("%d",x);
    return 0;
}
```

(A) 0 ○

(B) It will be difference of a and b ○

(C) It will be addition of a and b ○

(D) Compilation error ○

3

What will be output if you will execute following c code?

```
#include<stdio.h>
auto int a=5;
int main(){
    int x;
    x=~a+a&a+a<<a;
    printf("%d",x);
    return 0;
}
```

(A) 5 ○

(B) 0 ○

(C) 153 ○

(D) Compilation error ○

4

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    register int a,b;
    int c;
    scanf("%d%d",&a,&b);
    c=~a + ~b + ++a + b++;
    printf(" %d",c);
    return 0;
}
//User input is: 1 2
```

(A) -1 ○

(B) 0 ○

(C) 1 ○

(D) Compilation error ○

5

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int arr[3]={10,20,30};
    int x=0;
    x = ++arr[++x] + ++x + arr[--x];
    printf("%d ",x);
    return 0;
}
```

(A) 22 ○

(B) 23 ○

(C) 43 ○

(D) 44 ○

6

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int a[]={10,20,30,40};
    int i=3,x;
    x=1*a[--i]+2*a[--i]+3*a[--i];
    printf("%d",x);
    return 0;
}
```

(A) 30 ○

(B) 60 ○

(C) 90 ○

(D) Compilation error ○

7

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    static int a[][2][3]={0,1,2,3,4,5,6,7,8,9,10,11,12}
;
    int i=-1;
    int d;
    d=a[i++][++i][++i];
    printf("%d",d);
    return 0;
}
```

(A) 9 ○

(B) 10 ○

(C) 11 ○

(D) Compilation error ○

8

What will be output if you will execute following c code?

```
#include<stdio.h>
int f(int);
int main(){
    int i=3,val;
    val=sizeof (f(i)+ +f(i=1)+ +f(i-1));
    printf("%d %d",val,i);
    return 0;
}
int f(int num){
        return num*5;
}
```

(A) 2 3 ○

(B) 4 3 ○

(C) 3 2 ○

(D) Compilation error ○

9

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int x,a=3;
    x=+ +a+ + +a+ + +5;
    printf("%d  %d",x,a);
    return 0;
}
```

(A) 10 3 ○

(B) 11 3 ○

(C) 10 5 ○

(D) Compilation error ○

10

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int num,i=0;
    num=-++i+ ++-i;
    printf("%d",num);
    return 0;
}
```

(A) 0 ○

(B) 1 ○

(C) -2 ○

(D) Compilation error ○

11

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int num,a=5;
    num=-a--+ +++a;
    printf("%d  %d",num,a);
    return 0;
}
```

(A) 1 5 ○

(B) -1 6 ○

(C) 1 6 ○

(D) 0 5 ○

12

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int num,a=15;
    num=- - - -a--;
    printf("%d  %d",num,a);
    return 0;
}
```

(A) 15 14

(B) 14 15

(C) 14 14

(D) 15 15

13

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int x,a=2;
    x=++a,++a,a++;
    printf("%d  %d",x,a);
    return 0;
}
```

(A) 5 5

(B) 3 5

(C) 4 5

(D) 5 4

14

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int x,i=2;
    x=~-!++i;
    printf("%d",x);
    return 0;
}
```

(A) -2 ○
(B) -1 ○
(C) 0 ○
(D) 1 ○

15

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    static double *p,*q,*r,*s,t=5.0;
    double **arr[]={&p,&q,&r,&s};
    int i;
    *p=*q=*r=*s=t;
    for(i=0;i<4;i++)
        printf("%.0f  ",**arr[i]);
    return 0;
}
```

(A) 5 5 5 5 5 ○
(B) 5 6 7 8 9 ○
(C) Infinite loop ○
(D) Run time error ○

16

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    float x;
    x=0.35==3.5/10;
    printf("%f",x);
    return 0;
}
```

(A) 0.000000 ○

(B) 1.000000 ○

(C) 0.350000 ○

(D) Compilation error ○

17

```
#include<stdio.h>
int main(){
    int arr[]={6,12,18,24};
    int x=0;
    x=arr[1]+(arr[1]=2);
    printf("%d",x);
    return 0;
}
```

(A) 4 ○

(B) 8 ○

(C) 14 ○

(D) Compilation error ○

18

What will be output if you will execute following c code?

```
#include<stdio.h>
int sq(int);
int main(){
    int a=1,x;
    x=sq(++a)+sq(a++)+sq(a++);
    printf("%d",x);
     return 0;
}
int sq(int num){
    return num*num;
}
```

(A) 15 ○

(B) 16 ○

(C) 17 ○

(D) 18 ○

19

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    printf("%c",*"abcde");
     return 0;
}
```

(A) acbcd ○

(B) e ○

(C) a ○

(D) NULL ○

20

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    printf("%d","abcde"-"abcde");
     return 0;
}
```

(A) 0 ○

(B) -1 ○

(C) 1 ○

(D) Garbage ○

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int i;
    for(i=0;i<5;i++){
         int i=10;
         printf(" %d",i);
         i++;
    }
    return 0;
}
```

Wrong	10 11 12 13 14	○
Correct	10 10 10 10 10	○
(C)	0 1 2 3 4	○
(D)	Compilation error	○

Explanation:

Default storage class of local variable is auto. Scope of auto variables are block in which it has been declared. When program control goes out of the scope auto variables are dead. So variable *i*which has been declared inside for loop has scope within loop and in each iteration variable *i* is dead and re-initialized.
Note: If we have declared two variables of same name but different scope then local variable will have higher priority.

2

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    register a,b,x;
    scanf("%d %d",&a,&b);
    x=a+~b;
    printf("%d",x);
    return 0;
}
```

(A)	0	○
Wrong	It will be difference of a and b	○
(C)	It will be addition of a and b	○
Correct	Compilation error	○

Explanation:

Register variables are stored in CPU. So it has not memory address. Hence it is incorrect to write &a.

3

What will be output if you will execute following c code?

```
#include<stdio.h>
auto int a=5;
int main(){
    int x;
    x=~a+a&a+a<<a;
    printf("%d",x);
    return 0;
}
```

(A)	5	○
(B)	0	○
Wrong	153	○
Correct	Compilation error	○

Explanation:

We cannot declare auto variable outside of any function since it auto variables gets are created (i.e. gets memory) at run time.

4

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    register int a,b;
    int c;
    scanf("%d%d",&a,&b);
    c=~a + ~b + ++a + b++;
    printf(" %d",c);
    return 0;
}
//User input is: 1 2
```

(A)	-1	○
(B)	0	○
Wrong	1	○
Correct	Compilation error	○

Explanation:

Register variables are stored in CPU. So it has not memory address. Hence it is incorrect to write &a.

5

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int arr[3]={10,20,30};
    int x=0;
    x = ++arr[++x] + ++x + arr[--x];
    printf("%d ",x);
    return 0;
```

}

(A)	22	○
(B)	23	○
Correct	43	○
(D)	44	○

Explanation:

In Turbo C 3.0 and 4.5 compilers

Output: 43

Consider on expression:

```
= ++arr[++x] + ++x + arr[--x] //x = 0 + 1

= ++arr[++x] + ++x + arr[--x] //x = 1 + 1

= ++arr[++x] + ++x + arr[--x] //x = 2 - 1

= ++arr[1] + 1 + arr[1] //x = 1

= ++arr[1] + 1 + arr[1]  //arr[1] = 20+1

= arr[1] + 1 + arr[1] //arr[1] = 21

= 21 + 1 + 21

= 43
```

In Linux GCC complier

Output: 44

Consider on expression:

```
= ++arr[++x] + ++x + arr[--x] //x = 0 + 1

= ++arr[1] + ++x + arr[--x] ////x = 1 + 1

= ++arr[++x] + 2 + arr[--x] //x = 2 - 1

= ++arr[1] + 2 + arr[1] //arr[1] = 20+1

= arr[1] + 1 + arr[1] //arr[1] = 21
```

= 21 + 2 + 21

= 44

6

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int a[]={10,20,30,40};
    int i=3,x;
    x=1*a[--i]+2*a[--i]+3*a[--i];
    printf("%d",x);
    return 0;
}
```

(A)	30	○
Correct	60	○
Wrong	90	○
(D)	Compilation error	○

Explanation:

In Turbo C 3.0 and 4.5 compilers

Output: 60

```
Consider on expression:
= 1 * a[--i] + 2 * a[--i] + 3 * a[--i] //i = 3 - 2
= 1 * a[--i] + 2 * a[--i] + 3 * a[--i] //i = 2 - 1
= 1 * a[--i] + 2 * a[--i] + 3 * a[--i] //i = 1 - 1
= 1 * a[0] + 2 * a[0] + 3 * a[0] //i = 0
= 1 * 10 + 2 * 10 + 3 * 10 //a[0] = 10
= 10 + 20 + 30
= 60
```

In Linux GCC complier

Output: 90

```
Consider on expression:
= 1 * a[--i] + 2 * a[--i] + 3 * a[--i] //i = 3 - 2
= 1 * a[--i] + 2 * a[--i] + 3 * a[--i] //i = 2 - 1
= 1 * a[1] + 2 * a[1] + 3 * a[--i] //i = 1 - 1
= 1 * a[1] + 2 * a[1] + 3 * a[0]
= 1 * 20 + 2 * 20 + 3 * 10
= 20 + 40 + 30
= 90
```

7

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    static int a[][2][3]={0,1,2,3,4,5,6,7,8,9,10,11,12}
;
    int i=-1;
    int d;
    d=a[i++][++i][++i];
    printf("%d",d);
    return 0;
}
```

(A) 9 ○

Correct 10 ○

(C) 11 ○

Wrong Compilation error ○

Explanation:

```
= a[i++][++i][++i] //i = -1 +
1
= a[i++][++i][++i] //i = 0 + 1
= a[1][1][1] //i = 1 + 1
= 10
```

8

What will be output if you will execute following c code?

```
#include<stdio.h>
int f(int);
int main(){
    int i=3,val;
    val=sizeof (f(i)+ +f(i=1)+ +f(i-1));
    printf("%d %d",val,i);
    return 0;
}
int f(int num){
        return num*5;
}
```

Correct	2 3	○
Wrong	4 3	○
(C)	3 2	○
(D)	Compilation error	○

Explanation:

Turbo C 3.0 and Turbo C 4.5 compiler:
2 3

Linux GCC complier:

4 3

Any expression inside sizeof operator is never changed the value of the any variable. So value of variable *i* will remain 3. After the evaluation of expression inside sizeof operator we will get an integer value. So value of variable *val* will be sizeof int data type.

Note: Size of into in turbo C 3.0 and 4.5 is two byte while Linux gcc complier is four byte

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    int x,a=3;
    x=+ +a+ + +a+ + +5;
    printf("%d  %d",x,a);
    return 0;
}
```

(A)	10 3	○
Correct	11 3	○
Wrong	10 5	○
(D)	Compilation error	○

Explanation:

Consider on expression: + +a
Here both + are unary plus operation. So

```
= + +a+ + +a+ + +5;
= + +3+ + +3+ + 5
= 3+ 3+ 5
= 11
```

Note: Increment operator ++ cannot have space between two plus symbol.

10

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    int num,i=0;
    num=-++i+ ++-i;
    printf("%d",num);
    return 0;
}
```

(A)	0	○
(B)	1	○
(C)	-2	○
Correct	Compilation error	○

Explanation:

After operation of any operator on operand it returns constant value. Here we are performing unary minus operator on variable i so it will return a constant value and we can perform ++ operation on constant.

11

What will be output if you will execute following c code?

```c
#include<stdio.h>
int main(){
    int num,a=5;
    num=-a--+ +++a;
    printf("%d  %d",num,a);
    return 0;
}
```

(A)	1 5	○
(B)	-1 6	○
(C)	1 6	○
Correct	0 5	○

Explanation:

```
= -a--+ +++a
= -a-- + + ++a
= -a-- + + ++a
= -6 + + 6 //a = 6 -
1
= -6 + 6 //a = 5
= 0
```

12

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int num,a=15;
    num=- - - -a--;
    printf("%d  %d",num,a);
    return 0;
}
```

Correct 15 14 ○

(B) 14 15 ○

(C) 14 14 ○

Wrong 15 15 ○

Explanation:

```
= - - - -a
= - - - -15 //a = 15 – 1

= 15  //a = 14
```

13

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int x,a=2;
    x=++a,++a,a++;
    printf("%d  %d",x,a);
    return 0;
}
```

(A) 5 5 ○

Correct	3 5	○
(C)	4 5	○
(D)	5 4	○

Explanation:

```
x = ++a, ++a, a++
x = 3, ++a, a++ // a = 2 + 1
```

x **=** 3, ++a, a++ // = operator has higher precedence than comma operator

```
x = 3, ++a, a++ // a = 3 + 1

x = 3, 4, a++

x = 3, 4, 4 // a = 4 + 1

x = 3 // a = 5
```

14

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    int x,i=2;
    x=~-!++i;
    printf("%d",x);
    return 0;
}
```

(A)	-2	○
Correct	-1	○
(C)	0	○
(D)	1	○

Explanation:

```
= ~-!++i
= ~-!3 //i = 2 + 1
= ~-0 //!3 = 0
= ~0 //-0 = 0
= -(0 + 1) //~ is 1's complement operator.
= -1
```

15

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    static double *p,*q,*r,*s,t=5.0;
    double **arr[]={&p,&q,&r,&s};
    int i;
    *p=*q=*r=*s=t;
    for(i=0;i<4;i++)
        printf("%.0f  ",**arr[i]);
    return 0;
}
```

(A) 5 5 5 5 5 ○

(B) 5 6 7 8 9 ○

(C) Infinite loop ○

Correct Run time error ○

Explanation:

Turbo C 3.0:

5 5 5 5 5

Turbo C 4.5 and Linux GCC complier:

Run time error

16

What will be output if you will execute following c code?

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

```
int main(){
    float x;
    x=0.35==3.5/10;
    printf("%f",x);
    return 0;
}
```

Correct	0.000000	○
(B)	1.000000	○
Wrong	0.350000	○
(D)	Compilation error	○

Explanation:

Turbo C 3.0 and Turbo C 4.5 compiler:
Output: 0.000000
3.5/10 is little greater than .35

Linux GCC compilers:

Output: 1.000000

Note: == is logic operator. It returns 1 if both operands are equal otherwise it returns 0.

17

```
#include<stdio.h>
int main(){
    int arr[]={6,12,18,24};
    int x=0;
    x=arr[1]+(arr[1]=2);
    printf("%d",x);
    return 0;
}
```

Correct	4	○
Wrong	8	○
(C)	14	○
(D)	Compilation error	○

Explanation:

```
= arr[1] + (arr[1] = 2) //arr[i] = 2
= arr[1] + arr[1]

= 2 + 2

= 4
```

18

What will be output if you will execute following c code?

```c
#include<stdio.h>
int sq(int);
int main(){
    int a=1,x;
    x=sq(++a)+sq(a++)+sq(a++);
    printf("%d",x);
     return 0;
}
int sq(int num){
    return num*num;
}
```

(A)	15	○
Wrong	16	○
Correct	17	○
(D)	18	○

Explanation:

```
= sq(++a) + sq(a++) + sq(a++) //a= 1 + 1
= sq(2) + sq(2) + sq(a++) //a = 2 + 1
= sq(2) + sq(2) + sq(3)  //a = 3 + 1
= 4 + 4 + 9
= 17
```

Note: Pre-increment fist increment then assign while post increment operator first assign then increment.

19

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    printf("%c",*"abcde");
     return 0;
}
```

(A)	acbcd	○
(B)	e	○
Correct	a	○
(D)	NULL	○

Explanation:

String constant "abcde" will return memory address of first character of the string constant. *"abcde" will return the first character of string constant.

20

What will be output if you will execute following c code?

```
#include<stdio.h>
int main(){
    printf("%d","abcde"-"abcde");
     return 0;
}
```

(A)	0	○
(B)	-1	○
(C)	1	○

Correct Garbage

Explanation:

Memory address of string constants depends upon operating system.

Fail

It is **C programming language online test or quiz** objective types questions answers with explanation for interview or written test. If you have any queries or

Structure in c example

(q) What will be output of following c code?

```
void main()
{
    struct employee
    {
    unsigned id: 8;
    unsigned sex:1;
    unsigned age:7;
    };
    struct employee emp1={203,1,23};
    clrscr();
    printf("%d\t%d\t%d",emp1.id,emp1.sex,emp1.age);
    getch();
}
```

Output: 203 1 23
We can access the data member in same way.
How bit data is stored in the memory:

Minimum size of structure which data member in bit is two byte i.e. 16 bit. This is called word size of microprocessor. Word size depends on microprocessor. Turbo c is based on 8086 microprocessor which word size is two byte.

Bits are filed in from right to left direction 8 bit for id,1 bit for sex and 7 bit for age.

(q) What will be output of following c code?

```
void main()
{
    struct bitfield
    {
    unsigned a:5;
    unsigned c:5;
    unsigned b:6;

    }bit;
    char *p;
    struct bitfield *ptr,bit1={1,3,3};
    p=&bit1;
    p++;
    clrscr();
    printf("%d",*p);
    getch();
}
```

Output: 12
Explanation:
Binary value of a=1 is 00001 (in 5 bit)
Binary value of b=3 is 00011 (in 5 bit)
Binary value of c=3 is 000011 (in 6 bit)
In memory it is represented as:

Let address of bit1 is 500 which initialize to char pointer p. Since can is one byte data type so p++ will be 501. *p means content of memory location 501 which is (00001100) and its binary equivalent is 12. Hence output is 12.

(q) What will be output of following c code?

```
void main()
{
    struct bitfield
    {
    signed int a:3;
    unsigned int b:13;
    unsigned int c:1;
    };
    struct bitfield bit1={2,14,1};
    clrscr();
    printf("%d",sizeof(bit1));
    getch();
}
```

Output: 4
(q) What will be output of following c code?

```
void main()
{
    struct bitfield
    {
    unsigned a:3;
    char b;
    unsigned c:5;
    int d;
    }bit;
    clrscr();
    printf("%d",sizeof(bit));
    getch();
}
```

Output: 5

Note: (Actual output will 6 due to slack byte ,So Before executing this program first go to option menu then compiler then code generation then select word alignment then press OK)

(q) What will be output of following c code?

```
void main()
{
    struct field
    {
    int a;
    char b;
    }bit;
    struct field bit1={5,'A'};
    char *p=&bit1;
    *p=45;
    clrscr();
    printf("\n%d",bit1.a);
    getch();
}
```

Output: 45

Nesting of structure:

Nesting of structure is possible i.e. we can declare a structure within another structure but it is necessary inner structure must declares structure variable otherwise we can not access the data member of inner structure.

Example:

```
void main()
{
    struct world
    {
    int a;
    char b;
    struct india
    {
    char c;
    float d;
    }p;
    };
    struct world st ={1,'A','i',1.8};
    clrscr();
    printf("%d\t%c\t%c\t%f",st.a,st.b,st.p.c,st.p.d);
    getch();
}
```

```
Output: 1 A I 1.800000
```

(q) What will be output of following c code?

```
void main()
{
    struct india
    {
    char c;
    float d;
    };
    struct world
    {
    int a[3];
    char b;
    struct india orissa;
    };
    struct world st ={{1,2,3},'P','q',1.4};
    clrscr();
    printf("%d\t%c\t%c\t%f",st.a[1],st.b,st.orissa.c,st
.orissa.d);
    getch();
}
```

```
Output: 2 p q 1.400000
```

Printf function questions and answer with solution

Printf objective types interview questions and answers

(1)

```
#include<stdio.h>
#include<conio.h>
void main()
{
     int a=5,b=6,c=11;
     clrscr();
     printf("%d %d %d");
     getch();
}
```

What will output when you compile and run the above code?

(a)Garbage value garbage value garbage value
(b)5 6 11
(c)11 6 5
(d)Compiler error

Answer: (c)
(2)

```
#include<stdio.h>

void main()
{
     char *str="CQUESTIONBANK";
     clrscr();
     printf(str+9);
     getch();
}
```

What will output when you compile and run the above code?

(a)CQESTIONBANK

(b)CQUESTION
(c)BANK
(d)Compiler error

Answer: (c)
(3)

```
#include<stdio.h>
void main()
{
     clrscr();
     printf("%d",printf("CQUESTIONBANK"));
     getch();
}
```

What will output when you compile and run the above code?

(a)13CQUESTIONBANK
(b)CQUESTIONBANK13
(c)Garbage CQUESTIONBANK
(d)Compiler error

Answer: (b)

(4)
```
#include<stdio.h>
#include<conio.h>
void main()
{
     short int a=5;
     clrscr();
     printf("%d"+1,a);
     getch();
}
```

What will output when you compile and run the above code?
(a)6
(b)51

(c)d
(d)Compiler error

Answer: (c)

(5)
```
#include<stdio.h>
void main()
{
     int i=85;
     clrscr();
     printf("%p %Fp",i,i);
     getch();
}
```

What will output when you compile and run the above code?

(a)85 85
(b)0055 034E:0055
(c)0055 FFFF:0055
(d)Compiler error

Answer: (b)
(6)
```
#include<stdio.h>
static struct student
{
     int a;
     int b;
     int c;
     int d;
}s1={6,7,8,9},s2={4,3,2,1},s3;
void main()
{
     s3=s1+s2;
     clrscr();
     printf("%d %d %d %d",s3.a,s3.b,s3.c,s3.d);
     getch();
}
```

What will output when you compile and run the above code?
(a)6789
(b)4321
(c)10101010
(d)Compiler error

Answer: (d)

(7)
```
#include<stdio.h>
extern struct student
{
     int a;
     int b;
     int c;
     int d;
}s={6,7,8,9};

void main()
{
     clrscr();
     printf("%d %d %d %d",s.a,s.b,s.c,s.d);
     getch();
}
```

What will output when you compile and run the above code?
(a)6789
(b)9876
(c)0000
(d)Compiler error

Answer: (a)

(8)
```
#include<stdio.h>
struct student
{
     static int a;
```

```
    register int b;
    auto int c;
    extern int d;
}s={6,7,8,9};
void main()
{
    printf("%d %d % %d",s.a,s.b,s.c,s.d);
}
```

What will output when you compile and run the above code?
(a)6789
(b)9876
(c)0000
(d)Compiler error

Answer: (d)

(9)
```
#include<stdio.h>
struct student
{
    int roll;
    int cgpa;
    int sgpa[8];
};
void main()
{
    struct student s={12,8,7,2,5,9};
    int *ptr;
    ptr=(int *)&s;
    clrscr();
    printf("%d",*(ptr+3));
    getch();
}
```
What will output when you compile and run the above code?
(a)8
(b)7
(c)2

(d)Compiler error

Answer: (c)

(10)

```
#include<stdio.h>
struct game
{
     int level;
     int score;
     struct player
     {
     char *name;
     }g2={"anil"};
}g3={10,200};
void main()
{
     struct game g1=g3;
     clrscr();
     printf("%d  %d  %s",g1.level,g1.score,g1.g2.name);
     getch();
}
```

What will output when you compile and run the above code?

(a)10 200 anil
(b)200 10 anil
(c)10 200 null
(d)Compiler error

Answer: (d)

(11)

```
#include<stdio.h>
struct game
{
     int level;
     int score;
     struct player
     {
```

```
    char *name;
    }g2;
}g1;
void main()
{
    clrscr();
    printf("%d  %d  %s",g1.level,g1.score,g1.g2.name);
    getch();
}
```

What will output when you compile and run the above code?
(a)Garbage_value garbage_value garbage_value
(b)0 0 (null)
(c)Run time error
(d)Compiler error

Answer: (b)

String questions in c interview

Answer and explanation of questions are based on turbo c 3.0 compilers. Answer and explanation may vary in other compilers.

1.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    char arr[7]="Network";
    printf("%s",arr);
}
```

Choose all that apply:

(A) Network ○

(B) N ○

(C) Garbage value ○

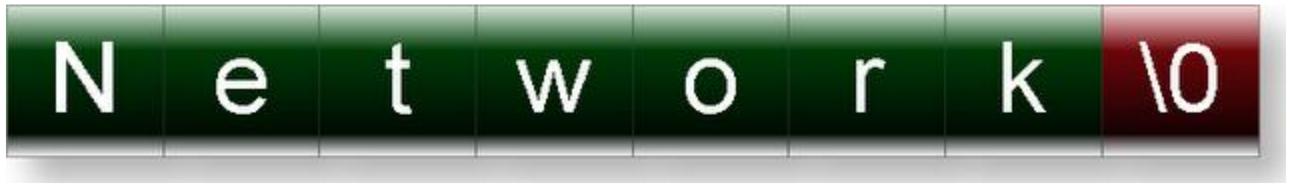
(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

Size of a character array should one greater than total number of characters in any string which it stores. In c every string has one terminating null character. This represents end of the string.



So in the string "Network" , there are 8 characters and they are 'N','e','t','w','o','r','k' and '\0'. Size of array arr is seven. So array arr will store only first sevent characters and it will note store null character.

As we know %s in prinf statement prints stream of characters until it doesn't get first null character. Since array arr has not stored any null character so it will print garbage value.

2.

WWhat will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    char arr[11]="The African Queen";
    printf("%s",arr);
}
```

Choose all that apply:

(A) The African Queen ○

(B) The

(C) The African

(D) Compilation error

(E) None of the above

Answer

Explanation:

Size of any character array cannot be less than the number of characters in any string which it has assigned. Size of an array can be equal (excluding null character) or greater than but never less than.

3.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int const SIZE=5;
    int expr;
    double value[SIZE]={2.0,4.0,6.0,8.0,10.0};
    expr=1|2|3|4;
    printf("%f",value[expr]);
}
```

Choose all that apply:

(A) 2.000000

(B) 4.000000

(C) 8.000000

(D) Compilation error

(E) None of the above

Answer

Explanation:

Size of any array in c cannot be constantan variable.

4.
What will be output when you will execute following c code?

```
#include<stdio.h>
enum power{
    Dalai,
    Vladimir=3,
    Barack,
    Hillary
};
void main(){
    float leader[Dalai+Hillary]={1.f,2.f,3.f,4.f,5.f};
    enum power p=Barack;
    printf("%0.f",leader[p>>1+1]);
}
```

Choose all that apply:

(A) 1 ○

(B) 2 ○

(C) 3 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

```
Size of an array can be enum constantan.
Value of enum constant Barack will equal to Vladimir +
1 = 3 +1 = 4
So, value of enum variable p  = 4
leader[p >> 1 +1]
= leader[4 >> 1+1]
=leader[4 >> 2]   //+ operator enjoy higher precedence
than >> operator.
=leader[1]  //4>>2 = (4 / (2^2) = 4/4 = 1
```

=2

5.

What will be output when you will execute following c code?

```c
#include<stdio.h>
#define var 3
void main(){
    char *cricket[var+~0]={"clarke","kallis"};
    char *ptr=cricket[1+~0];
    printf("%c",*++ptr);
}
```

Choose all that apply:

(A) a ○

(B) r ○

(C) l ○

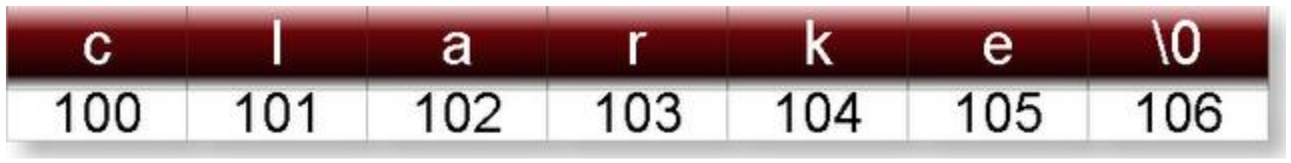
(D) Compilation error ○

(E) None of the above ○

Answer

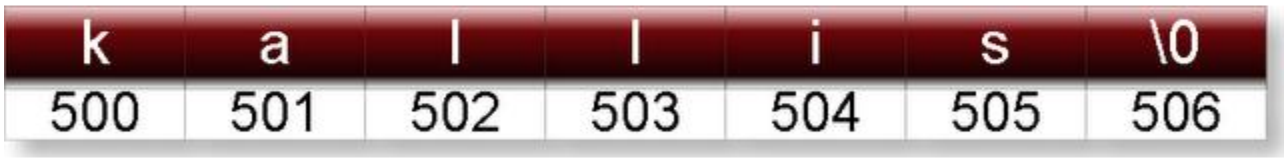
Explanation:

In the expression of size of an array can have micro constant.
var +~0 = 3 + ~0 = 3 + (-1) = 2
Let's assume string "clarke" and "kallis" has stored at memory address 100 and 500 respectively as shown in the following figure:
For string "clarke":



c	l	a	r	k	e	\0
100	101	102	103	104	105	106

For string "kallis":



k	a	l	l	i	s	\0
500	501	502	503	504	505	506

In this program cricket is array of character's pointer of size 2. So array cricket will keep the memory address of first character of both strings i.e. content of array cricket is:
cricket[2] = {100,500}
ptr is character pointer which is pointing to the fist element of array cricket. So, ptr = 100
Now consider on *++ptr
Since ptr = 100 so after ++ptr , ptr = 101
*(++ptr) = *(101) = content of memory address 101. From above figure it is clear that character is l.

6.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
    char data[2][3][2]={0,1,2,3,4,5,6,7,8,9,10,11};
    printf("%o",data[0][2][1]);
}
```

Choose all that apply:

(A) 5

(B) 6

(C) 7

(D) Compilation error

(E) None of the abov

Answer

Explanation:

%o in printf statement is used to print number in the octal format.

7.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    short num[3][2]={3,6,9,12,15,18};
    printf("%d  %d",*(num+1)[1],**(num+2));
}
```

Choose all that apply:

(A) 12 18 ○

(B) 18 18 ○

(C) 15 15 ○

(D) 12 15 ○

(E) Compilation error ○

Answer

Explanation:

```
*(num+1)[1]
=*(*((num+1)+1))
=*(*(num+2))
=*(num[2])
=num[2][0]
=15
And
**(num+2)
=*(num[2]+0)
=num[2][0]
=15
```

8.

What will be output when you will execute following c code?

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

```
void main(){
    char *ptr="cquestionbank";
    printf("%d",-3[ptr]);
}
```

Choose all that apply:

(A) Garbage value ○

(B) -300 ○

(C) -101 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

```
-3[ptr]
=-*(3+ptr)
=-*(ptr+3)
=-ptr[3]
=-103  //ASCII value of character 'e' is 103
```

9.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    long  myarr[2][4]={0l,1l,2l,3l,4l,5l,6l,7l};
    printf("%ld\t",myarr[1][2]);
    printf("%ld%ld\t",*(myarr[1]+3),3[myarr[1]]);

printf("%ld%ld%ld\t" ,*(*(myarr+1)+2),*(1[myarr]+2),3[1
[myarr]]);
}
```

Choose all that apply:

(A) 5 66 776 ○

(B) 6 77 667

(C) 6 66 776

(D) Compilation error

(E) None of the above

Answer

Explanation:

Think yourself.

10.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int array[2][3]={5,10,15,20,25,30};
    int (*ptr)[2][3]=&array;
    printf("%d\t",***ptr);
    printf("%d\t",***(ptr+1));
    printf("%d\t",**(*ptr+1));
    printf("%d\t",*(*(*ptr+1)+2));
}
```

Choose all that apply:

(A) 5 Garbage 20 30

(B) 10 15 30 20

(C) 5 15 20 30

(D) Compilation error

(E) None of the above

Answer

Explanation:

ptr is pointer to two dimension array.
***ptr

```
=***&array  //ptr = &array
=**array //* and & always cancel to each other
=*arr[0]  // *array = *(array +0) = array[0]
=array[0][0]
= 5
Rests think yourself.
```

11.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    static int a=2,b=4,c=8;
    static int *arr1[2]={&a,&b};
    static int *arr2[2]={&b,&c};
    int* (*arr[2])[2]={&arr1,&arr2};
    printf("%d %d\t",*(*arr[0])[1],
*(*(**(arr+1)+1)));
}
```

Choose all that apply:

(A) 2 4 ○

(B) 2 8 ○

(C) 4 2 ○

(D) 4 8 ○

(E) None of the above ○

Answer

Explanation:

```
Consider on the following expression:
*(*arr[0])[1]
=*(*&arr1)[1]  //arr[0] = &arr1
=*arr1[1]   //* and & always cancel to each other
=*&b
=b
```

```
=4
Consider on following expression:
*(*(**(arr+1)+1))
= *(*(*arr[1]+1))  //*(arr+1) = arr[1]
= *(*(*&arr2+1))  //arr[1] = &arr2
=*(*(arr2+1))  //*&arr2 = arr2
=*(arr2[1])  //*(arr2+1) = arr2[1]
=  *&c    //arr2[1] = &c
=  c
= 8
```

12.

What will be output when you will execute following c code?

```
#include<stdio.h>
#include<math.h>
double myfun(double);
void main(){
    double(*array[3])(double);
    array[0]=exp;
    array[1]=sqrt;
    array[2]=myfun;

printf("%.1f\t",(*array)((*array[2])((**(array+1))(4)))
);
}
double myfun(double d){
       d-=1;
       return d;
}
```

Choose all that apply:

(A) 3.5 ○

(B) 7.0 ○

(C) 2.7 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

array is array of pointer to such function which parameter is double type data and return type is double.

```
Consider on following expression:
(*array)((*array[2])((**(array+1))(4)))
= (*array)((*array[2])((*array[1])(4)))
//*(array+1) = array[1]
= (*array)((*array[2])(sqrt(4))))
//array[1] = address of sqrt function
= (*array)((*array[2])(2.000000)))
= (*array)(myfun(2.000000)))
// array[2] = address of myfunc function
=(*array)(1.000000)
=array[0](1.000000)
=exp(1.000000)
```

13.

What will be output when you will execute following c code?

```c
#include<stdio.h>
typedef struct{
    char *name;
    double salary;
}job;
void main(){
    static job a={"TCS",15000.0};
    static job b={"IBM",25000.0};
    static job c={"Google",35000.0};
    int x=5;
    job * arr[3]={&a,&b,&c};
    printf("%s  %f\t",(3,x>>5-4)[*arr]);
}
double myfun(double d){
       d-=1;
       return d;
}
```

Choose all that apply:

(A) TCS 15000.000000 ○

(B) IBM 25000.000000 ○

(C) Google 35000.000000 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

```
(3,5>>5-4)[*arr]
=(3,5>>5-4)[*arr] //x=5
= (3,5>>1)[*arr] //- operator enjoy higher precedence
than >>
= (3,2)[*arr]  //5>>1 = 5/(2^1) = 5 /2 = 2
= 2[*arr]  //In c comma is also operator.
= *(2 + *arr)
= *(*arr + 2)
=*arr[2]
=*(&c) //arr[2] = &c
=c   // *  and & always cancel to each other.
So,
printf("%s  %f\t",c);
=> printf("%s  %f\t", "Google",35000.0);
```

14.

What will be output when you will execute following c code?

```
#include<stdio.h>
union group{
    char xarr[2][2];
    char yarr[4];
};
void main(){
    union group x={'A','B','C','D'};
    printf("%c",x.xarr[x.yarr[2]-67][x.yarr[3]-67]);
}
```

Choose all that apply:

(A) A ○

(B) B ○

(C) C ○

(D) D ○

(E) Compilation error ○

Answer

Explanation:

In union all member variables share common memory space.

So union member variable, array xarray will look like:

```
{
{'A','B'},
{'C','D'}
}
```

And union member variable, array yarray will look like:

```
{
{'A','B','C','D'}
}
x.xarr[x.yarr[2]-67][x.yarr[3]-67]
= x.xarr['C'-67]['D'-67]
= x.xarr[67-67][68-67]
//ASCII value of 'C' is 67 and 'D' is 68.
x.xarr[0][1]
='B'
```

15.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int a=5,b=10,c=15;
    int *arr[3]={&a,&b,&c};
    printf("%d",*arr[*arr[1]-8]);
}
```

Choose all that apply:

(A) 5 ○

(B) 10 ○

(C) 15 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

Member of an array cannot be address of auto variable because array gets memory at load time while auto variable gets memory at run time.

16.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int arr[][3]={{1,2},{3,4,5},{5}};
    printf("%d %d %d",sizeof(arr),arr[0][2],arr[1][2]);
}
```

Choose all that apply:

(A) 12 3 5 ○

(B) 18 0 5 ○

(C) 12 0 5 ○

(D) 18 3 5 ○

(E) Compilation error ○

Answer

Explanation:

If we will not write size of first member of any array at the time of declaration then size of the first dimension is max elements in the initialization of array of that dimension.
So, size of first dimension in above question is 3.
So size of array = (size of int) * (total number of elements) = 2 *(3*3) = 18
Default initial value of rest elements are zero. So above array will look like:

```
{
{1,2,0}
{3,4,5},
{5,0,0}
}
```

17.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int xxx[10]={5};
    printf("%d %d",xxx[1],xxx[9]);
}
```

Choose all that apply:

(A) Garbage Garbage ○

(B) 0 0 ○

(C) null null ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

If we initialize any array at the time of declaration the compiler will treat such array as static variable and its default value of uninitialized member is zero.

18.

What will be output when you will execute following c code?

```
#include<stdio.h>
#define WWW -1
enum {cat,rat};
void main(){
    int Dhoni[]={2,'b',0x3,01001,'\x1d','\111',rat,WWW}
;
    int i;
    for(i=0;i<8;i++)
         printf(" %d",Dhoni[i]);
}
```

Choose all that apply:

(A) 2 98 4 513 28 73 1 -1 ○

(B) 2 98 4 507 29 73 2 -1 ○

(C) 2 99 3 513 29 73 2 -1 ○

(D) 2 98 3 513 29 73 1 -1 ○

(E) Compilation error ○

Answer

Explanation:

Dhoni[0]=2
Dhoni[1]='b' =98 //ASCII value of character 'b' is 98.
Dhoni[2]= 0x3 = 3 //0x represents hexadecimal number. Decimal value of hexadecimal 3 is also 3.
Dhoni[3]=01001 = 513 //Number begins with 0 represents octal number.
Dhoni[4] = '\x1d' = 29 //'\x1d' is hexadecimal character constant.

```
Dhoni[5] = '\111' = 73 //'\111' is octal character constant.
Dhoni[6] =rat = 1  //rat is enum constant
Dhoni[7] = WWW = -1  //WWW is macro constant.
```

19.

What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    long double a;
    signed char b;
    int arr[sizeof(!a+b)];
    printf("%d",sizeof(arr));
}
```

Choose all that apply:

(A) 16 ○

(B) 4 ○

(C) 2 ○

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

Size of data type in TURBO C 3.0 compiler is:

S.N.	Data type	Size(In byte)
1	char	1
2	int	2
3	double	8

Consider on the expression: !a + b
! Operator always return zero if a is non-zero number other wisie 1. In general we can say ! operator always returns an int type number. So
!a +b

=! (Any double type number) + Any character type number
= Any integer type number + any character type number
= Any integer type number
Note: In any expression lower type data is always automatically type casted into the higher data type. In this case char data type is automatically type casted into the int type data.
So sizeof (!a +b) = sizeof(Any int type number) = 2
So size of array arr is 2 and its data type is int. So sizeof(arr) = size of array * sizeof its data type = 2* 2= 4

20.

What will be output when you will execute following c code?

```c
#include<stdio.h>
void main(){
    char array[]="Ashfaq \0 Kayani";
    char *str="Ashfaq \0 Kayani";
    printf("%s %c\n",array,array[2]);
    printf("%s %c\n",str,str[2]);
    printf("%d %d\n",sizeof(array),sizeof(str));
}
```

Choose all that apply:

(A)
```
Ashfaq h
Ashfaq h
16 2
```
○

(B)
```
Ashfaq h
Ashfaq Kayani h
16 16
```
○

(C)
```
Ashfaq y
Ashfaq h
```
○

2 2

(D) Compilation error ○

(E) None of the above ○

Answer

Explanation:

A character array keeps the each element of an assigned array but a character pointer always keeps the memory address of first element.
As we know %s in prints the characters of stream until it doesn't any null character ('\0'). So first and second printf function will print same thing in the above program. But size of array is total numbers of its elements i.e. 16 byte (including ending null character). While size of any type of pointer is 2 byte (near pointer).

String questions with solution of c programming

C pointers interview questions and answers

Frequently asked technical objective types multiple choice pointer questions with explanation of placement in c programming language
Note: Linux GCC compilers and Visual C++ compiler doesn't support far and huge pointers.

1.
What will be output of following program?

```c
#include<stdio.h>
int main(){
   int a = 320;
   char *ptr;
   ptr =( char *)&a;
   printf("%d ",*ptr);
   return 0;
}
```

(A) 2 ○

(B) 320 ○

(C) 64 ○

(D) Compilation error ○

(E) None of above ○

Output

Explanation:

Turbo C++ 3.0: 64

Turbo C ++4.5: 64

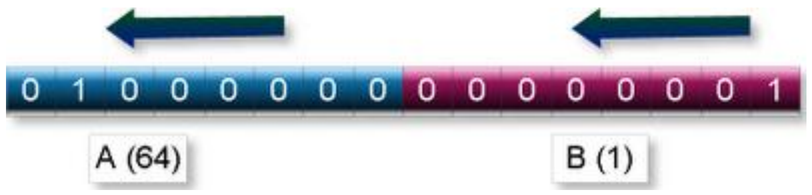
Linux GCC: 64

Visual C++: 64

As we know int is two byte data byte while char is one byte data byte. char pointer can keep the address one byte at time.

Binary value of 320 is 00000001 01000000 (In 16 bit)

Memory representation of int a = 320 is:



So ptr is pointing only first 8 bit which color is green and Decimal value is 64.

2.
What will be output of following program?

```
#include<stdio.h>
#include<conio.h>
int main(){
   void (*p)();
   int (*q)();
   int (*r)();
   p = clrscr;
   q = getch;
   r = puts;
  (*p)();
  (*r)("cquestionbank.blogspot.com");
  (*q)();
  return 0;
}
```

(A) NULL ○
(B) cquestionbank.blogspot.com ○
(C) c ○
(D) Compilation error ○
(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: cquestionbank.blogspot.com

Turbo C ++4.5: cquestionbank.blogspot.com

Linux GCC: Compilation error

Visual C++: Compilation error

p is pointer to function whose parameter is void and return type is also void. r and q is pointer to function whose parameter is void and return type is int . So they can hold the address of such function.

3.

What will be output of following program?

```
#include<stdio.h>
int main(){
   int i = 3;
   int *j;
   int **k;
   j=&i;
   k=&j;
   printf("%u %u %d ",k,*k,**k);
   return 0;
}
```

(A) Address, Address, 3 ○

(B) Address, 3, 3 ○

(C) 3, 3, 3 ○

(D) Compilation error ○

(E) None of above ○

Output

Explanation:

Turbo C++ 3.0: Address, Address, 3

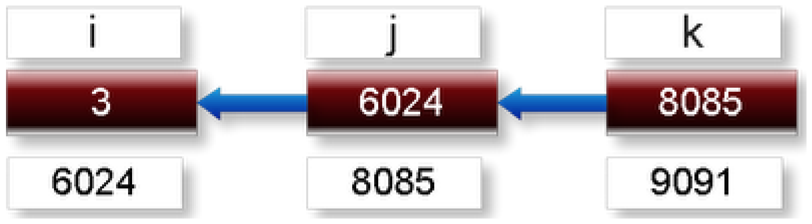
Turbo C ++4.5: Address, Address, Address

Linux GCC: Address, Address, 3

4.

Visual C++: Address, Address, 3

Memory representation



Here 6024, 8085, 9091 is any arbitrary address, it may be different.

Value of k is content of k in memory which is 8085

Value of *k means content of memory location which address k keeps.

k keeps address 8085 .

Content of at memory location 8085 is 6024

In the same way **k will equal to 3.

Short cut way to calculate:

Rule: * and & always cancel to each other

i.e. *&a = a

So *k = *(&j) since k = &j

*&j = j = 6024

And
**k = **(&j) = *(*&j) = *j = *(&i) = *&i = i = 3

4.

What will be output of following program?

```
#include<stdio.h>
int main(){
   char far *p =(char far *)0x55550005;
   char far *q =(char far *)0x53332225;
   *p = 80;
   (*p)++;
   printf("%d",*q);
   return 0;
}
```

(A) 80

(B) 81

(C) 82

(D) Compilation error

(E) None of above

Output

Explanation:

Turbo C++ 3.0: 81

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Far address of p and q are representing same physical address.

Physical address of 0x55550005 = (0x5555) * (0x10) + (0x0005) = 0x55555

Physical address of 0x53332225 = (0x5333 * 0x10) + (0x2225) = 0x55555 *p = 80, means content at memory location 0x55555 is assigning value 25

(*p)++ means increase the content by one at memory location 0x5555 so now content at memory location 0x55555 is 81

*q also means content at memory location 0x55555 which is 26

5.

What will be output of following program?

```
#include<stdio.h>
#include<string.h>
int main(){
    char *ptr1 = NULL;
    char *ptr2 = 0;
    strcpy(ptr1," c");
    strcpy(ptr2,"questions");
    printf("\n%s %s",ptr1,ptr2);
    return 0;
}
```

(A) c questions

(B) c (null)

(C) (null) (null)

(D) Compilation error

(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0: (null) (null)

Turbo C ++4.5: Run time error

Linux GCC: Run time error

Visual C++: Run time error

We cannot assign any string constant in null pointer by strcpy function.

6.

What will be output of following program?

```
#include<stdio.h>
int main(){
     int huge *a =(int huge *)0x59990005;
     int huge *b =(int huge *)0x59980015;
     if(a == b)
     printf("power of pointer");
     else
     printf("power of c");
     return 0;
}
```

(A) power of pointer ○
(B) power of c ○
(C) power of cpower of c ○
(D) Compilation error ○
(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: power of pointer

Turbo C ++4.5: power of c

Linux GCC: Compilation error

Visual C++: Compilation error

Here we are performing relational operation between two huge addresses. So first of all both a and b will normalize as:

a= (0x5999)* (0x10) + (0x0005) =0x9990+0x0005=0x9995

b= (0x5998)* (0x10) + (0x0015) =0x9980+0x0015=0x9995

Here both huge addresses are representing same physical address. So a==b is true.

7.

What will be output of following program?

```
#include<stdio.h>
#include<string.h>
int main(){
     register a = 25;
     int far *p;
     p=&a;
     printf("%d ",*p);
     return 0;
}
```

(A) 25 ○
(B) 4 ○
(C) Address ○
(D) Compilation error ○
(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: Compilation error

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Register data type stores in CPU. So it has not any memory address. Hence we cannot write &a.

8.

What will be output of following program?

```
#include<stdio.h>
#include<string.h>
int main(){
    char far *p,*q;
    printf("%d %d",sizeof(p),sizeof(q));
    return 0;
}
```

(A) 2 2 ○

(B) 4 4 ○

(C) 4 2 ○

(D) 2 4 ○

(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 4 4

Turbo C ++4.5: 4 4

Linux GCC: Compilation error

Visual C++: Compilation error

p is far pointer which size is 4 byte.

By default q is near pointer which size is 2 byte.

9.

What will be output of following program?

```
#include<stdio.h>
int main(){
    int a = 10;
    void *p = &a;
    int *ptr = p;
    printf("%u",*ptr);
    return 0;
}
```

(A) 10 ○

(B) Address ○

(C) 2 ○

(D) Compilation error ○

(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 10

Turbo C ++4.5: 10

Linux GCC: 10

Visual C++: 10

Void pointer can hold address of any data type without type casting. Any pointer can hold void pointer without type casting.

10.

What will be output of following program?

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

```
#include<string.h>
int main(){
    int register a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
//if a=25
```

(A) 25
(B) Address
(C) 0
(D) Compilation error
(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0: Compilation error

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Register data type stores in CPU. So it has not any memory address. Hence we cannot write &a.

11.

What will be output of following program?

```
#include<stdio.h>
int main(){
    char arr[10];
    arr = "world";
    printf("%s",arr);
    return 0;
}
```

(A) world ○

(B) w ○

(C) Null ○

(D) Compilation error ○

(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: Compilation error

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Compilation error Lvalue required

Array name is constant pointer and we cannot assign any value in constant data type after declaration.

12.

What will be output of following program?

```
#include<stdio.h>
#include<string.h>
int main(){
    int a,b,c,d;
    char *p = ( char *)0;
    int *q = ( int *q)0;
    float *r = ( float *)0;
    double *s = 0;
    a = (int)(p+1);
    b = (int)(q+1);
    c = (int)(r+1);
```

```
     d = (int)(s+1);
     printf("%d %d %d %d",a,b,c,d);
return 0;
}
```

(A) 2 2 2 2

(B) 1 2 4 8

(C) 1 2 2 4

(D) Compilation error

(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 1 2 4 8

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Address + 1 = next address

Since initial address of all data type is zero. So its

next address will be size of data type.

13.

What will be output of following program?

```
#include<stdio.h>
#include<string.h>
int main(){
     int a = 5,b = 10,c;
     int *p = &a,*q = &b;
```

```
    c = p - q;
    printf("%d" , c);
    return 0;
}
```

(A) 1 ○

(B) 5 ○

(C) -5 ○

(D) Compilation error ○

(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 1

Turbo C ++4.5: 1

Linux GCC: 1

Visual C++: 2

Difference of two same type of pointer is always one.

14.
What will be output of following program?

```
#include<stdio.h>
unsigned long int (* avg())[3]{
    static unsigned long int arr[3] = {1,2,3};
    return &arr;
}
int main(){
    unsigned long int (*ptr)[3];
    ptr = avg();
    printf("%d" , *(*ptr+2));
    return 0;
}
```

(A) 1

(B) 2

(C) 3

(D) Compilation error

(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 3

Turbo C ++4.5: 3

Linux GCC: 3

Visual C++: 3

15.

What will be output of following program?

```
#include<stdio.h>
int main(){
    int * p , b;
    b = sizeof(p);
    printf("%d" , b);
return 0;
}
```

(A) 2

(B) 4

(C) 8

(D) Compilation error

(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 2 or 4

Turbo C ++4.5: 2 or 4

Linux GCC: 4

Visual C++: 4

since in this question it has not written p is which type of pointer. So its output will depend upon which memory model has selected. Default memory model is small.

16.

What will be output of following program?

```
#include<stdio.h>
int main(){
    int i = 5 , j;
    int *p , *q;
    p = &i;
    q = &j;
    j = 5;
    printf("%d %d",*p,*q);
    return 0;
}
```

(A) 5 5 ○

(B) Address Address ○

(C) 5 Address ○

(D) Compilation error ○

(E) None of above ○

Output

Explanation:

Turbo C++ 3.0: 5 5

Turbo C ++4.5: 5 5

Linux GCC: 5 5

Visual C++: 5 5

17.

What will be output of following program?

```c
#include<stdio.h>
int main(){
    int i = 5;
    int *p;
    p = &i;
    printf(" %u %u", *&p , &*p);
    return 0;
}
```

(A) 5 Address ○

(B) Address Address ○

(C) Address 5 ○

(D) Compilation error ○

(E) None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: Address Address

Turbo C ++4.5: Address Address

Linux GCC: Address Address

Visual C++: Address Address

Since * and & always cancel to each other.

i.e. *&a = a

so *&p = p which store address of integer i

&*p = &*(&i) //since p = &i

= &(*&i)

= &i

So second output is also address of i

18.

What will be output of following program?

```
#include<stdio.h>
int main(){
    int i = 100;
    printf("value of i : %d addresss of i : %u",i,&i);
    i++;
    printf("\nvalue of i : %d addresss of i :
%u",i,&i);
    return 0;
}
```

(A)
```
value of i : 100 addresss of i : Address
value of i : 101 addresss of i : Address
```
(B)
```
value of i : 100 addresss of i : Address
value of i : 100 addresss of i : Address
```
(C)
```
value of i : 101 addresss of i : Address
value of i : 101 addresss of i : Address
```
(D) Compilation error

(E) None of above

Output

E x p l a n a t i o n :

Turbo C++ 3.0:

```
value of i : 100 addresss of i : Address

value of i : 101 addresss of i : Address

Turbo C ++4.5:

value of i : 100 addresss of i : Address

value of i : 101 addresss of i : Address

Linux GCC:

value of i : 100 addresss of i : Address

value of i : 101 addresss of i : Address

Visual C++:

value of i : 100 addresss of i : Address
value of i : 101 addresss of i : Address
```

Within the scope of any variable, value of variable may change but its address will never change in any modification of variable.

19.

What will be output of following program?

```
#include<stdio.h>
int main(){
    char far *p =(char far *)0x55550005;
    char far *q =(char far *)0x53332225;
    *p = 25;
    (*p)++;
    printf("%d",*q);
    return 0;
}
```

(A) 25 ○

(B) Address ○

(C) Garbage ○
(D) Compilation error ○
(E)None of above ○

Output

E x p l a n a t i o n :

Turbo C++ 3.0: 26

Turbo C ++4.5: Compilation error

Linux GCC: Compilation error

Visual C++: Compilation error

Far address of p and q are representing same physical address. Physical address of

0x55550005 = 0x5555 * ox10 + ox0005 = 0x55555

Physical address of

0x53332225 = 0x5333 * 0x10 + ox2225 = 0x55555

*p = 25, means content at memory location 0x55555 is assigning value 25

(*p)++ means to increase the content by one at memory the location 0x5555 so now content of memory location at 0x55555 is 26

*q also means content at memory location 0x55555 which is 26

20.

What will be output of following program?

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

```
int main(){
    int i = 3;
    int *j;
    int **k;
    j = &i;
    k = &j;
    printf("%u %u %u",i,j,k);
return 0;
}
```

(A) 3 Address 3 ○

(B) 3 Address Address ○

(C) 3 3 3 ○

(D) Compilation error ○

(E) None of above ○

Output

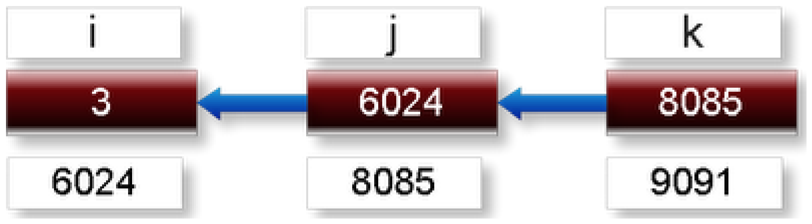
Explanation:

Turbo C++ 3.0: 3 Address Address

Turbo C ++4.5: 3 Address Address

Linux GCC: 3 Address Address

Visual C++: 3 Address Address



Here 6024, 8085, 9091 is any arbitrary address, it may be different.