Pointers

Find out the output for the following programs.

1) #include <stdio.h >
```
int main()
{
    int *p = 10;
    printf(“ %u\n”, (unsigned int)p);
    printf(“%d\n”,*p);
}
```
Ans..10,error

2) #include <stdio.h>
```
int main()
{
  int *ptr, a = 10;
  ptr = &a;
  *ptr += 1;
  printf("%d,%d/n", *ptr, a);

}
```
Ans...11,11

3) #include<stdio.h>
```
int main()
{
    int x = -300;
    unsigned char *p;
    p = &x;
    printf(“%d\n”,*p++);
    printf(“%d\n”,*p);
}
```
Ans..212,254

4) #include<stdio.h>
```
 int main()
 {
    int x = 256;
    char *p = &x;
    *++p = 2;
    printf(“%d”,x);
 }
```
Ans..512

5) #include<stdio.h>
```
int main()
{
    int x = 300;
    if(*(char *)&x == 44)
    printf(“Little Endian\n”);
    else
```

```
        printf(“Big Endian\n”);
    }
```

Ans..little endian

6)
```
#include <stdio.h>
void main()
{
    int x = 0;
    int *ptr = &5;
    printf("%p\n", ptr);
}
```

Ans..lvalue request error as & is unary operand

7)
```
#include<stdio.h>
int main()
{
        int const *p = 5;
        int q;
        p = &q;
        printf(“%d”,++(*p));
}
```

Ans..read only permission for *p

8)
```
#include<stdio.h>
int main()
{
        int x = 10;
        int const * const p;
        p = &x;
        printf(“%d\n”, *p);
}
```

Ans..read only variable p

9)
```
#include <stdio.h>
int x = 0;
void main()
{
    int *const ptr = &x;
    printf("%p\n", ptr);
    ptr++;
    printf("%p\n ", ptr);
}
```

Ans..read only variable ptr

10)
```
#include <stdio.h>
int main()
 {
    const int ary[4] = {1, 2, 3, 4};
    int *p;
    p = ary + 3;
    *p = 5;
```

```
        printf("%d\n", ary[3]);
    }
```
Ans...5

```
11) #include <stdio.h>
    int main()
    {
        int ary[4] = {1, 2, 3, 4};
        int *p = ary + 3;
        printf("%d\n", p[-2]);    }
```
Ans..2

```
12)  #include <stdio.h>
    void main()
    {
        char *s= "hello";
        char *p = s + 2;
        printf("%c\t%c", *p, s[1]);
    }
```
Ans..l e

```
13) #include <stdio.h>
    int main()
    {
        void *p;
        int a[4] = {1, 2, 3, 4};
        p = &a[3];
        int *ptr = &a[2];
        int n = (int*)p - ptr;
        printf("%d\n", n);
    }
```
Ans...1

```
14) #include<stdio.h>
   int main()
   {
       int a[ ] = {10,20,30,40,50},i;
       char *p = a;

       for(i=0;i<5;i++)
      printf("%d   ",*p++);
   }
```
Ans...10,0,0,0,20

```
15)  #include<stdio.h>
    int main()
    {
        int a[]={10,20,30,40,50};
        char *p;
        p=(char *)a;
        printf("%d\n",*((int *)p+4));
    }
```

Ans..50

16) #include <stdio.h>
```
int main()
{
   double *ptr = (double *)100;
   ptr = ptr + 2;
   printf("%u\n", ptr);
}
```
Ans..116

17) #include <stdio.h>
```
int main()
{
   int i = 10;
   void *p = &i;
   printf("%d\n", (int *)*p);
// printf("%d\n", *(int*)p);
   return 0;
}
```
Ans..error due to invalid use of void exp

18) #include <stdio.h>
```
int main()
{
   int a[4] = {1, 2, 3, 4};
   void *p = &a[1];
   void *ptr = &a[2];
   int n = 1;
   n = ptr - p;
   printf("%d\n", n);
}
```
Ans....4

19) #include <stdio.h>
```
int main()
{
   int *p = (int *)2;
   int *q = (int *)3;
   printf("%d", p + q);
}
```
Ans...because + is not possible with pointers.

20) Which of the following operand can be applied to pointers p and q?
(Assuming initialization as int *a = (int *)2; int *b = (int *)3;)

a) a + b
b) a – b
c) a * b
d) a / b

Ans: b)

21) Which of following logical operation can be applied to pointers?
(Assuming initialization int *a = 2; int *b = 3;)

a) a | b
b) a ^ b
c) a & b
d) None of the mentioned

Ans: d)

22)
```
#include <stdio.h>
void main()
{
   char *s = "hello";
   char *n = "cjn";
   char *p = s + n;
    4  printf("%c\t%c", *p, s[1]);
}
```
Ans...error because + operand is not possible with pointers

23)
```
#include <stdio.h>
void m(int *p)
{
   int i = 0;
   for(i = 0;i < 5; i++)
   printf("%d\t", p[i]);
}
void main()
{
   int a[5] = {6, 5, 3};
   m(&a);
}
```
Ans..6 5 3 0 0

24)
```
#include <stdio.h>
void foo(int*);
int main()
{
   int i = 10,j=20,*p = &i;
   foo(p++);
      foo(p);
}
void foo(int *p)
{
   printf("%d\n", *p);
}
```
Ans..10,20

25)#include <stdio.h>
```
int main()
{
    int i = 97, *p = &i;
    foo(&i);
    printf("%d ", *p);
}
void foo(int *p)
{
    int j = 2;
    p = &j;
    printf("%d ", *p);
}
```
Ans..2,97

26) #include<stdio.h>
```
int main()
{
    const int ary[4] = {1,2,3,4};
    int *p = ary+3;
    *p = 5;
    ary[3] = 6;
    printf("%d",ary[3]);
}
```
Ans..read only permission for array

27) #include<stdio.h>
```
int main()
{
    char *p = "Hai friends",  *p1 = p;
    while(*p!='\0');
    ++*p++;
    printf("%s  %s\n",p,p1);
}
```
Ans...infinite loop

28) #include<stdio.h>
```
int main()
{
    char *x = "VECTOR";
    printf("%s\n",x+3);
    printf("%d\n"+1,123456);
}
```
Ans..TOR,d

29) #include<stdio.h>
```
int main()
{
    char a[ ] = "abcdefgh";
```

```
        int *ptr = a;
        printf("%x  %x\n",ptr[0],ptr[1]);
    }
```
Ans...64636461,64636465

30)
```
#include<stdio.h>
#include<string.h>
int main()
{
    char *str = "hello, world\n";
    char *strc = "good morning\n";
    strcpy(strc, str);
    printf("%s\n", strc);
    return 0;
}
```
Ans.. segmentation fault

31)
```
#include <stdio.h>
int main()
{
    char *str = "hello world";
    char strc[50] = "good morning india\n";
    strcpy(strc, str);
    printf("%s\n", strc);
    return 0;
}
```
Ans..hello world

32)
```
#include <stdio.h>
int main()
{
    char *str = "hello, world\n";
    str[5] = '.';
    printf("%s\n", str);
    return 0;
}
```
Ans.. segmentation fault

33)
```
#include <stdio.h>
int main()
{
    char str[] = "hello, world";
    str[5] = '.';
    printf("%s\n", str);
    return 0;
}
```
Ans .hello. world

34)
```
#include <stdio.h>
int main()
{
    char *str = "hello world";
```

```
    char strary[] = "hello world";
    printf("%d %d\n", sizeof(str), sizeof(strary));
    return 0;
}
```

Ans.4,12

35)
```
#include <stdio.h>
int main()
{
    char *str = "hello world";
    char strary[] = "hello world";
    printf("%d %d\n", strlen(str), strlen(strary));
    return 0;
}
```

Ans.11,11

36)
```
#include<stdio.h>
int main()
{
    int a = 5,b = 4,c = 9;
    *(a>b ? &a : &b) = (a+b)>c;
    printf("%d  %d\n",a,b);
}
```

Ans..0,4

37) Find the sizeof any datatype with out using sizeof operator. (Hint : Use pointers)

38)
```
#include<stdio.h>
int main()
{
    int i;
    double a = 5.2;
    char *ptr;
    ptr = (char *)&a;
    for(i=0;i<=7;i++)
    printf("%d\n",*ptr++);
    return 0;
}
```

Ans....-51
-52
-52r
-52
-52
-52
20
64

39) Correct the following program.

```
#include<stdio.h>
int main()
{
void *p;
        int  **ptr;
```

```
        int a = 129;
    p = &a;
        ptr = &p;
    printf(“ p = %d   p = %u  &p = %u\n”, *p, p, &p);
    }
```

Ans...by explicit type casting of p

40)
```
#include<stdio.h>
main()
{
    char a[20];
    char *p,*q;
    p=&a[0];
    q=&a[10];
    printf("%d %d\n",q-p,&q-&p);
}
```

Ans...10,1

41)
```
#include<stdio.h>
main()
{
        int a=0x12345678;
        void *ptr;
        ptr=&a;
        printf("0x%x\n",*(int *)&*&*(char*)ptr);
}
```

Ans...0x12345678

42)
```
#include<stdio.h>
main()
{
    int a[5]={1,2,3,4,5};
    int *ptr=(int *)(&a+1);
    printf("%d %d\n",*(a+1),*(ptr-1));
    printf("%d %d\n",*(a+1),*(ptr));
}
```

Ans..2 ,5
2,-1078301204

43)
```
#include <stdio.h>
    void main()
    {
      char *s= "hello";
      char *p = s;
      printf("%c\t%c", 1[p], s[1]);
    }
```

Ans..e e

44)
```
#include<stdio.h>
main()
```

```
{
    char a[]="abcde";
    char *p=a;
    p++;
    p++;
    p[2]='z';
    printf("%s",p);
}
```

Ans...cdz

45)
```
#include<stdio.h>
main()
 {
char a[]=”ABCDEFGHIJKLMNOPQRSTUVWXYZ”;
        int i,*p = a;
        for(i=0;i<5;i++)
        printf(“%d\t”,*p++);
 }
```

Ans...1145258551 1212630597 1280002633 1347374669 1414746705 v

46)
```
#include<stdio.h>
  main()
  {
char a[]=”abcdef”;
char *ptr1 = a;
ptr1 = ptr1+(strlen(ptr1)-1);
printf(“%c”, --*ptr1--);
printf(“%c”,--*--ptr1);
printf(“%c”,--*(ptr1--));
printf(“%c”,--*(--ptr1));
printf(“%c”,*ptr1);
}
```

Ans..ecbaa

47)
```
#include<stdio.h>
int main()
{
char *str1 = “Hello”;
char *str2 = “Hai”;
char *str3;
str3 = strcat(str1,str2);
printf(“%s  %s\n”,str3,str1);
return 0;
}
```

Ans... segmentation fault

48)
```
#include<stdio.h>
int main()
{
char a[]=”Hello”;
char *p=”Hai”;
a=”Hai”;
```

```
p=”Hello”;

printf(“%s  %s\n”,a,p);
return 0;
}
```

Ans...error due to incompatible types when assigning bwn pointer and array

49)
```
#include<stdio.h>
int main()
{
int i,n;
char *x=”Alice”;
n=strlen(x);
*x=x[n];
for(i=0;i<=n;i++)
{
        printf(“%s”,x);
        x++;
}
printf(“%s\n”,x);
return 0;
}
```

ANS.. Segmentation fault

50)
```
#include<stdio.h>
char *str=”char *str=%c%s%c;main(){printf(str,34,str,34);}”;
int main()
{
printf(str,34,str,34);
return 0;
}
```

Ans...char *str=”char *str=%c%s%c;main(){printf(str,34,str,34);}";main(){printf(str,34,str,34);}

51)
```
#include <stdio.h>
void f(char *k)
{
   k++;
   k[2] = 'm';
   printf("%c\n", *k);
}
void main()
{
   char s[] = "hello";
   f(s);
   printf("%s\n",s);
}
```

Ans...e helmo

52)
```
#include<stdio.h>
void t1(char *q);
main()
{
```

```
    char *p;
    p = “abcder”;
    t1(p);
}
void t1(char *q)
{
    if(*q!='r')
    {
            putchar(*q);
            t1(q++);
    }
}
```

Ans.....aaaaaaaaaaaaaaa.. segmentation fault

53)
```
#include<stdio.h>
int main(){
int i;
float a=5.2;
char *ptr;
ptr=(char *)&a;
for(i=0;i<=3;i++)
printf("%d ",*ptr++);
return 0;
}
```

Ans..102 102 -90 64

54)
```
#include <stdio.h>
void foo( int[] );
int main()
{
    int ary[4] = {1, 2, 3, 4};
    foo(ary);
    printf("%d ", ary[0]);
}
void foo(int p[4])
{
    int i = 10;
    p = &i;
    printf("%d ", p[0]);
}
```

Ans..10 1

55)
```
#include <stdio.h>
void main()
{
    int k = 5;
    int *p = &k;
    int **m  = &p;
     **m = 10;
    printf("%d%d%d\n", k, *p, **m);
}
```

Ans..10 10 10

56) #include <stdio.h>
```
int main()
{
   int a = 1, b = 2, c = 3;
   int *ptr1 = &a, *ptr2 = &b, *ptr3 = &c;
   int **sptr = &ptr1;
      printf("%d  ",**sptr);
   *sptr = ptr2;
      printf(("%d  ",**sptr);
}
```
Ans..some error

57) #include <stdio.h>
```
void main()
{
   int a[3] = {1, 2, 3};
   int *p = a;
   int *r = &p;
   printf("%d\n", (**r));

}
```
Ans..error.. invalid type of argument of unary *

58) #include <stdio.h>
```
int main()
{
   int i = 97, *p = &i;
   foo(&p);
   printf("%d ", *p);
   return 0;
}
void foo(int **p)
{
   int j = 2;
   *p = &j;
   printf("%d ", **p);
}
```
Ans...2,2

59) #include <stdio.h>
```
void foo(int *const *p);
int main()
{
   int i = 11;
   int *p = &i;
   foo(&p);
   printf("%d ", *p);
}
void foo(int *const *p)
{
```

```
    int j = 10;
    *p = &j;
    printf("%d ", **p);
}
```

Ans...read only permission

60)
```
#include <stdio.h>
void foo(int **const p);
int main()
{
    int i = 10;
    int *p = &i;
    foo(&p);
    printf("%d ", *p);
}
void foo(int **const p)
{
    int j = 11;
    *p = &j;
    printf("%d ", **p);
}
```

Ans..11,11

61)
```
#include <stdio.h>
int *f();
int main()
{
     int *p = f();
     printf("%d\n", *p);
}
int *f()
{
     int *j = (int*)malloc(sizeof(int));
    *j = 10;
     return j;

}
```

Ans..10

62)
```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
    int i = 0;
    for (i = 0;i < 10; i++)
    printf("%s  ", *(a[i]));
}
```

Ans.. segmentation fault

63)
```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
```

```
    int i = 0, j = 0;
    a[0] = "hey";
    for (i = 0;i < 10; i++)
    printf("%s  ", a[i]);
}
```
Ans...hey hello how null null null null null null null

64)
```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
    printf("%d\n", sizeof(a));
}
```
Ans..40

65)
```
#include <stdio.h>
void main()
{
    char *a[10] = {"hi", "hello", "how"};
    printf("%d\n", sizeof(a[1]));
}
```
Ans..4

66)
```
#include <stdio.h>
int main()
{
    char a[2][6] = {"hello", "hi"};
    printf("%s  ", *a + 1);
    return 0;
}
```
Ans..ello

67)
```
#include <stdio.h>
int main()
{
    char *a[2] = {"hello", "hi"};
    printf("%s\n", *(a + 1));
    return 0;
}
```
Ans..hi

68)
```
#include <stdio.h>
int main(int argc, char *argv[])
{
    while (argc--)
    printf("%s\n", argv[argc]);
    return 0;
}
```

69)
```
#include <stdio.h>
int main(int argc, char *argv[])
```

```
{
   while (*argv++ != NULL)
   printf("%s\n", *argv);
   return 0;
}
```

70)
```
#include <stdio.h>
int main(int argc, char *argv[])
{
   while (*argv  !=  NULL)
   printf("%s\n", *(argv++));

  return 0;
}
```

71)
```
#include<stdio.h>
int main(int sizeofargv, char *argv[])
{
while(sizeofargv)
printf("%s  ",argv[--sizeofargv]);
return 0;
} if i/p is   sample  friday tuesday sunday
```

Ans....a.out

72)
```
#include<stdio.h>
int main()
{
char *str[]={"Progs","Do","Not","Die","They","Croak!"};
printf("%d  %d",sizeof(str),strlen(str[0]));
return 0;
}
```

Ans...24 5

73)
```
#include<stdio.h>
int main()
{
static char *s[]={"black","white","pink","violet"};
char **ptr[]={s+3,s+2,s+1,s},***p;
p = ptr;
printf("%s\n",**p+1);
return 0;
}
```

Ans..iolet

74)
```
#include<stdio.h>
main()
{
char *m[]={"jan","feb","mar"};
char d[][10] = {"sun","mon","tue"};
printf("%s\t",m[1]);
printf("%s\t",d[1]);
}
```

Ans..Feb mon

75)
```
#include<stdio.h>
void fun(char **);
int main()
{
char *argv[]={“ab”,”cd”,”ef”,”gh”};
fun(argv);
return 0;
}
void fun(char **p)
{
char *t;
t=(p+=sizeof(int))[-1];
printf(“%s\n”,t);
}
```

Ans...gh

76)
```
#include <stdio.h>
void first()
{
   printf("first");
}
void second()
{
   first();
}
void third()
{
   second();
}
void main()
{
   void (*ptr)();
   ptr = third;
   ptr();
}
```

Ans..first

77)
```
#include <stdio.h>
int add(int a, int b)
{
   return a + b;
}
int main()
{
   int (*fn_ptr)(int, int);
   fn_ptr = add;
   printf("The sum of two numbers is: %d\n", (int)fn_ptr(2, 3));
}
```

Ans...the sum of two numbers is 5

78)
```
#include <stdio.h>
int mul(int a, int b, int c)
```

```
{
   return a * b * c;
}
void main()
{
   int (*function_pointer)(int, int, int);
   function_pointer  =  mul;
   printf("The product of three numbers is:%d",
   function_pointer(2, 3, 4));
}
```

Ans..the product of numbers is 24

79)
```
#include<stdio.h>
int fun(int (*)());
int main()
{
fun(main);
printf("Hi\n");
return 0;
}
int fun(int (*p)())
{
printf("Hello\n");
return 0;
}
```

Ans...Hello
Hi

80)
```
#include<stdio.h>
int main()
{
  char *p = "Hello World";
  printf(p);
  }
```

Ans..hello world

---------- END ----------

Dear Students, if any mistakes found, Kindly inform to me.

A.Tandava Ramakrishna
Email:ramakrishna@vectorindia.org