

1. Find the output for the following programs(branching and looping)

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

```
Void main()
```

```
{  
  int i;  
  for( i = 1 ; i < 4 ; i++)  
  {  
    switch(i)  
    {  
      case 1 : printf("%d" , i);break;  
      case 2 : printf("%d" , i);break;  
      case 3 : printf("%d" , i);break;  
    }  
  
  }  
  switch(i)  
  {  
    case 4 : printf("%d" , i);break;  
  }  
}
```

Output : 1234

2. Find the output(operator and expression)

```
void main()  
{  
  char *s = "\12345s\n";  
  printf("%d" , sizeof(s));  
}
```

Output : 4

3. Find the output(Functions)

```
int main()  
{  
  static int i = 3;  
  printf("%d" , i--);  
  return i>0 ? main() : 0 ;  
}
```

Output : 321

4. Find the output(pointers)

```
int main()  
{
```

```
char *s[]={ "dharmr'a","hewlett-packard","siemens","ibm"};
char **p;
p = s ;
```

```
printf("%s" ,++*p);
printf("%s",*p++); ;
printf("%s" ,++*p);
}
```

Output: harmr'aharmr'aewlett-packard

5. Find the output(dynamic memory)

```
#include<stdio.h>
#include<malloc.h>
#include<string.h>
int main()
{
int i;
char a[]="String";
char *p = "New String";
char *temp;
temp = malloc(strlen(p) + 1);
p = malloc( strlen(temp) + 1);
strcpy(p , temp);
printf("%s" , p);
}
```

Output : unpredictable string

6. Find the output(algorithm)

```
int main()
{
int n = 12 , res = 1;
while( n > 3)
{
n -= 3;
res *= 3;
}
printf("%d" , n*res);
}
```

Output : 81

7. Find the output(function)

```
void fun(int[][3]);
int main()
```

```

{
int a[3][3] = {9,8,7,6,5,4,3,2,1};
fun(a);
printf("%d\n" , a[2][1]);
}
void fun(int b[][3])
{
    ++b;
    b[1][1]=5;
}

```

Output : 5

8. Find the output(strings)

```

void main()
{

    int i , n;
    char x[5];
    strcpy( x , "Zoho");
    n = strlen(x);
    *x = *(x+(n-1));
    printf("%s" , x);
}

```

Output: ooho

9. Find the output(arrays)

```

void main()
{
    int c[]={5,4,3,4,5};
    int j , *q = c;
    for( j = 0 ; j<5 ; j++){
        printf("%d" , *c);
        ++q;
    }
}

```

Output:55555

10. Find the output(branching and looping)

```

void main()
{
    int i = 1;
    for(i=0 ; i= -1 ; i=1){
        printf("%d", i);
    }
}

```

```

        if(i!= 1) break;
    }
}

```

Output: -1

11. Find the output(Arrays)

```

void main()
{

    int s[] = { 1,0,5,0,10,0};
    int f[] = {2,4,6,8,10,12};
    int n = 6 , i = 0 , j = 0;
    for( j = 1 ; j < n ; j++)
    {
        if( s[j] >= f[i])
        {
            printf("%d" , i);
            i = j;
        }
    }
}

```

output : 02

12. Find the output(Functions)

```

void f(int *a , int m)
{
    int j = 0;
    for(j = 0 ; j < m ; j++)
    {
        *(a+j) = *(a+j) - 5;
    }
}

void main()
{
    int a[] = { 'f' , 'g' , 'h' , 'i' , 'j' };
    int j = 0 ;
    f(a , 5);
    for(j = 0 ; j <= 4 ; j++)
        printf("%c\t" , a[j]);

}

```

Output:a b c d e

13. Find the output(branching and looping)

```
void main()
{
    int i=0, j=0 , sum=0;
    for(i= 1; i < 500 ; i*=3)
        for(j=0;j<i;j++)
            sum++;
    printf("%d",sum);

}
```

Output: 364

14. Find the output(branching and looping)

```
void main()

{

    int n;

    for(n = 6 ; n!= 1; n--)

        printf("%d" , n--);

}
```

Output: infinite loop

15. Find the output(arrays)

```
void main()
{
    int a[3][4] = {2,4,6,5,10,12,12,10,5,6,4,2};
    int i = 0 , j , k =99;
    while(i < 3)
    {
        for(j = 0 ; j < 4 ; j= j++)
        {
            if( a[i][j] < k)
            {
                k = a[i][j];
            }
        }
        i++;
    }
}
```

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

Output : 2

16. Find the output(pointer)

```
void main()
{

    char *x="Alice";
    int i , n = strlen(x);
    *x = x[n];
    for(i=0; i<=n; i++)
    {
        printf("%s ", x); x++;
        printf("\n", x);
    }

    return 0;
```

```
}
```

Output : runtime error

17. Find the output(structures and union)

```
struct value{
    int bit1:1;
    int bit3:4;
    int bit4:4;
}bit;
int main()
{
    printf("%d\n", sizeof(bit));
    return 0;
}
```

Output : 4

18. Find the output(dynamic memory)

```
struct node
{
    int data;
    float d;
    struct node *link;
};
```

```

int main()
{
    struct node *p, *q;
    p = (struct node *) malloc(sizeof(struct node));
    q = (struct node *) malloc(sizeof(struct node));
    printf("%d, %d\n", sizeof(p), sizeof(q));
    return 0;
}

```

Output : 4 , 4

19. Find the output(structures and unions)

```

typedef union
{
    int a;
    char b[10];
    float c;
}Union;
int main()
{
    Union x , y = { 100};
    x.a = 50;
    strcpy(x.b , "Hello");
    x.c = 21.50;
    printf("%d %s %f\n" , x.a , x.b , x.c);
    printf("%d %s %f" , y.a,y.b, y.c);

}

```

Output:1101791232 21.500000

100 d 0.000000

20. Find the output(structures and union)

```

struct point{

    int x;
    int y ;
};
struct point origin , *pp;
int main()
{
    pp = &origin;
    printf("origin is (%d %d)\n" , (*pp).x , (*pp).y);
    printf("origin is (%d %d)" , pp->x , pp->y);
}

```

```
    return 0;
}
```

Output : origin is (0 0)

origin is (0 0)

21. Find the output(branching and looping)

```
void main()
{
    int i = -1;
    printf("i=%d +i = %d\n" , i , +1);
}
```

Output : i=-1 i=1

22. Find the output(datatypes)

```
void main()
{
    char not;
    not=12;
    printf("%d",not);

}
```

Output : 12

23. Find the output(branching and looping)

```
#define FALSE -1
#define TRUE 1
#define NULL 0
void main()
{
    if(NULL)
        puts("NULL");
    else if(FALSE)
        puts("TRUE");
    else
        puts(" FALSE");

}
```

Output : TRUE

24. Find the output(operator and expressions)

```
void main()
{
    int k = 1;
```



```
    printf("%d==1 is"" %s",k, k == 1 ? "TRUE":"FALSE");
}
```

Output : 1==1 is TRUE

25. Find the output(file manipulation)

```
int main()
{
    FILE *ptr;
    char i;
    ptr=fopen("demo.c","r");
    while((i=fgetch(ptr))!=EOF)
    printf("%c",i);
}
```

26. Find the output(branching and looping)

```
int main()
{
    int t , i ;
    for ( t=4;scanf("%d",&i)-t;printf("%d\n",i))
        printf("%d--",t--);

}
```

Output : loop runs 4 times

27. Find the output(structures and unions)

```
struct emp{
    int len;
    char name[1];
};
int main()
{
    char newname[] = "Rahul";
    struct emp *p = (struct emp *) malloc(sizeof(struct emp) * 1 + strlen(newname)+
    1);
    p->len = strlen(newname);
    strcpy(p -> name, newname);
    printf("%d %s\n", p->len, p->name); return 0;
}
```

Output : 5 Rahul

28. Find the output(algorithm)

```
int main() {
    printf("%d %d %d %d\n",72,072,0x72,0X72);
    return 0;
```

```
}
```

Output : 72 58 114 114

29. Find the output(operator and expression)

```
void main()
{
    char ch;
    int a;
    float b;
    printf("bytes occupied by ch=%d\n",sizeof(ch));
    printf("bytes occupied by a=%d\n",sizeof(a));
    printf("bytes occupied by b=%d\n",sizeof(b));
}
```

Output :

Bytes occupied by ch=1

Bytes occupied by a=4

Bytes occupied by b=4

30. Find the output(operator and expressions)

```
void main()
{
    printf("%d\n" , sizeof('7'));
    printf("%d\n" , sizeof(7));
    printf("%d\n" , sizeof(7.0));
}
```

Output: 4

4

8

31. Find the output(datatypes)

```
void main()
{
    char ch=291;
    printf("%d %d %c\n",2147483648,ch,ch);
    return 0;
}
```

Output : -2147483648 35 #

32. Find the output(datatypes)

```
void main()
{
    int g;
    g=300000*300000/300000;
    printf("g=%d\n",g);
}
```

```
}
```

Output : -647

33. Find the output(datatypes)

```
void main()
```

```
{
```

```
    float a;
```

```
    a=4/2;
```

```
    printf("%f %f\n",a,4/2);
```

```
}
```

Output : 2.000000 0.000000

34. Find the output(operator and expression)

```
void main()
```

```
{
```

```
    printf("%d\n",sizeof(4)/sizeof(2.0));
```

```
    printf("%d\n",sizeof(2.0)/sizeof(4));
```

```
}
```

Output : 0 2

35. Find the output(operator and expression)

```
void main()
```

```
{
```

```
    int x=10,y=5,p,q;
```

```
    p=x > 9;
```

```
    q=x>3&& y!=3;
```

```
    printf("p=%d q=%d \n",p,q);
```

```
}
```

Output : p = 1 q=1

36. Find the output