

1. Find the output of the below program

```
void fun(int a[],int size)
{
    int k = 0,i,j,data[6],num=0;
    for(i=1,j=0;i<size-1;i=i+1)
    {
        if(a[i]<a[i-1] && a[i+1]<a[i])
        {
            data[j]=a[i];
            j=j+1;
            if(j!=1)
            {
                if(a[i]>data[j-2]){
                    num=data[j-1];
                }
            }
            else{num=data[j-1];}
        }
    }
    printf("%d ",num);
}

int main()
{
    int array[5]={5,4,3,-3,1};
    fun(array,5);
}
```

2. Find the output of the below program

```
int main()
{
    int mat[2][2]={2,4,6,1} ,N=2;
    for (int x = 0; x < N / 2; x=x+1)
    {
        for (int y = x; y < N-x-1; y=y+1)
        {
            int temp = mat[x][y];
            mat[x][y] = mat[y][N-1-x];
            mat[y][N-1-x] = mat[N-1-x][N-1-y];
            mat[N-1-x][N-1-y] = mat[N-1-y][x];
            mat[N-1-y][x] = temp;
        }
    }
    for(int i=0;i<N;i=i+1)
    {
        for(int j=0;j<N;j=j+1)
        {
            printf("%d*",mat[i][j]);
        }
    }
}
```

3. Find the output of the below program

```
int main()
{
    int n=5, X=65;
    int A[] = {10, 20, 30, 40, 5, 15};
    for (int i = 0; i < n-3; i=i+1)
    {
        for (int j = i+1; j < n-2; j=j+1)
        {
            for (int k = j+1; k < n-1; k=k+1)
            {
                for (int l = k+1; l < n; l=l+1)
                {
                    if (A[i] + A[j] + A[k] + A[l] == X)
                        printf("%d,%d,%d,%d", A[i], A[j], A[k], A[l]);
                }
            }
        }
    }
}
```

4. Find the output of the below program

```
int main()
{
    int i, j;
    int p, q;
    int arr[] = {-7, 1, 5, 2, -4, 3, 0}, n=7;
    for ( i = 0; i < n; i=i+1)
    {
        p = 0;
        q = 0;

        for ( j = 0; j < i; j=j+1)
            p = p+ arr[j];

        for( j = i+1; j < n; j=j+1)
            q = q+ arr[j];

        if (p == q)
            printf("%d*", i);
    }
    printf("hello");
}
```

5. Find x in output

Note: Do not write the entire sequence. Enter only the value of 'x'.

```
int main()
{
    int mat[3][3] = {3,1,7,5,4,7,2,9,3},temp=0,i,j;
    for (j=0;j<3;j=j+1)
    {
        for (i = 0; i < 3; i=i+1)
        {
            temp = temp + mat[i][j];
        }
        printf("%d ",temp);
        temp = 0;
    }
}
```

Output: 'x' 14 17

6. Find 'x' in the below program to get the desired output.

```
int main()
{
    int n = 'x',i;
    for(i=2;i<n/2;i=i+1)
    {
        if(n%i==0)
            printf("%d ",i);
    }
}
```

Output: 2 3 5 6 10

7. Find 'x' in the below program to get the desired output.

```
int main()
{
    int c, n = 'x', result = 1;

    for(c = 1; c < n; c=c+1)
    {
        if(c%2==0 || c==7)
        {
            result = result*c;
        }
    }
    printf("%d",result);
}
```

Output: 48

8. Find the output of the following program.

```
void fun(int n)
{
    if(n > 1)
    {
        n=n-1;
        fun(n-1);
        printf("%d*", n);
    }
}
int main()
{
    int a=6;
    fun(a);
}
```

9. Find x in output

```
int main()
{
    int count=0;
    for (int i=1; i<=4;i=i+1)
    {
        for (int j=1;j<=4;j=j+1)
        {
            count=count+1;
            j=j+1;

            for (int k=0;k<=j;k=k+1)
            {
                count=count+1;
                printf("%d ",count);
            }
            if (j<=3){break;}
        }
    }
}
```

Output: 2 3 5 6 7 8 10 11 'x' 14 15 16

10. Find the output of the given program.

```
int main()
{
    int n = 5, result = 0;
    for(int i=0; i<n; i=i+1)
    {
        if(i<n/2)
        {
            result = result - i;
        }
        else
        {
            result = result + i;
        }
    }
    printf("%d", result);
    return 0;
}
```

11. Find the output of the given program.

```
int main()
{
    int a[] = {3, 2, 1, 6, 5}, i, j, temp;
    for(i=0; i<5; i=i+1)
    {
        for(j=0; j<5; j=j+1)
        {
            if((a[i]%2)<(a[j]%2))
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    for(i=0; i<5; i=i+1)
    {
        printf("%d*", a[i]);
    }
    return 0;
}
```

12. Find the output of the below program. Hint: '%' operator returns remainder of integer division.(Eg: 17 % 5 = 2)

```
int main()
{
    int a[3]={5,9,1};
    for(int i=0;i<3;i=i+1)
    {
        if(a[i]%3==0)
        {
            printf("%d#",a[i]*3);
        }
        else if(a[i]%2==1)
        {
            printf("%d#",a[i]*12);
        }
        if(a[i]%2!=0)
        {
            printf("%d#",a[i]+a[i]);
        }
    }
}
```

13. Find the output of the given program.

```
int main()
{
    int n=50, i, result = 0;
    for(i=1; i<=n; i=i+1)
    {
        if(i%2==0)
        {
            result = result + 1;
        }
        if(i%4==0)
        {
            result = result - 2;
        }
        if(i%8==0)
        {
            result = result + 1;
        }
    }
    printf("%d", result);
}
```

14. Find the output of the given program.

```
void main()
{
    int x=4,y=10,m,a,b,t;
    a = x;
    b = y;
    while (b != 0)
    {
        t = b;
        b = a % b;
        a = t;
    }
    m = (x*y)/a;
    printf("%d,%d",m,a);
}
```

15. Find the output of the below program:

```
int one(int a)
{
    printf("A");
    int d=a+1;
    return d;
}
int two(int b)
{
    printf("B");
    int f=b+one(b);
    return f;
}
int three(int c)
{
    printf("C");
    int z=c+one(c)+two(c);
    return z;
}
int maze(int x)
{
    int res = x+three(x);
    return res;
}
int main()
{
    int y = maze(5);
    printf("%d",y);
}
```

16. Find the output of the below program.

```
int main()
{
    for(int x=0;x<=3;x=x+1)
    {
        switch(x){
            case 3: printf("R");
            default: printf("Z");
        }
    }
}
```


17. Find the output of the below program.

```
void compute(int m)
{
    int n=m,i=1,j=0,k=0;
    while(n>0)
    {
        n=n/10;    i=i*10;
    }
    i=i/10;
    while(i>0)
    {
        j=(m/i);
        if(j%3==0)
        {
            k=(k*10)+1;
        }
        else
        {
            k=k*10;
        }
        i=i/10;
    }
    printf("%d",k);
}
int main()
{
    compute(97892);
}
```

18. Find X in output of the below program:

Output: 3 6 9 90 49 X

```
int main()
{
    int i;
    for(i=0;i<20;i=i+1)
    {
        do
        {
            printf("%d ",i=i+3);
        }while(i<8);
        switch(i)
        {
            case 9: printf("%d ",i+(i*i));
            case 10: printf("%d ",((i*i)/2)+i);
                     break;
            default: printf("%d ",i);
        }
        i=i+1;
        if(i>15)
        {
            break;
        }
        else
        {
            printf("%d ",i=i+(i*2));
        }
    }
}
```



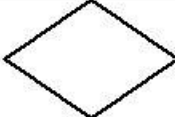

19. Find the output of the below program.

```
int main()
{
    int n = 4;
    while (n >= 2)
    {
        for (int c = 1; c < n; c++)
        {
            if ((n+c)%2 == 0)
                printf("#");
            else
                printf("$");
        }
        n = n - 2;
    }
}
```

20. Find the output of the below program.

```
int main()
{
    int i=0,j=0,k=0,m=0,n=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if((i+j)%2==0)
            {
                m=m+1;
            }
            n=n+1;
        }
        for(k=0;k<3;k++)
        {
            if((i+k)%2==0)
            {
                n=n+1;
            }
            m=m+1;
        }
    }
    printf("%d",n+m);
}
```

Description of the notations used in the flow charts:

Symbol	Description
	Denotes the beginning or end of a flowchart.
	Denotes a process to be carried out.
	Denotes a decision to be made. (Similar to IF...ELSE... statement in a programming language)
	Denotes the direction of logic flow.

Instructions for the flow charts:

- The input for some of the flow-charts would be stored in a memory location. The values for the memory location can be accessed by enclosing the variable inside the brackets [].

Example

Address:	100	101	102	103
Value:	6	5	2	8

N1 = 100

VAL = [N1]

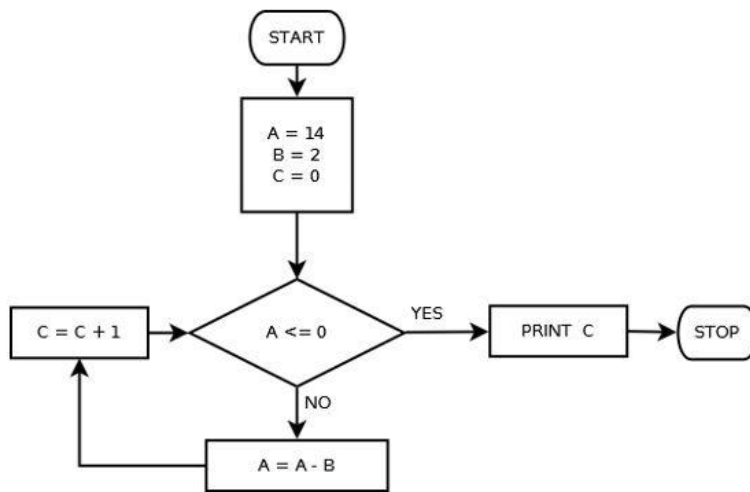
In the above sample, the value that is stored in memory location 100 has been accessed as '[N1]'. So, the 'VAL' variable will have 6.

- And for the 'Find the output' questions, it is enough to write the output for PRINT statements alone.

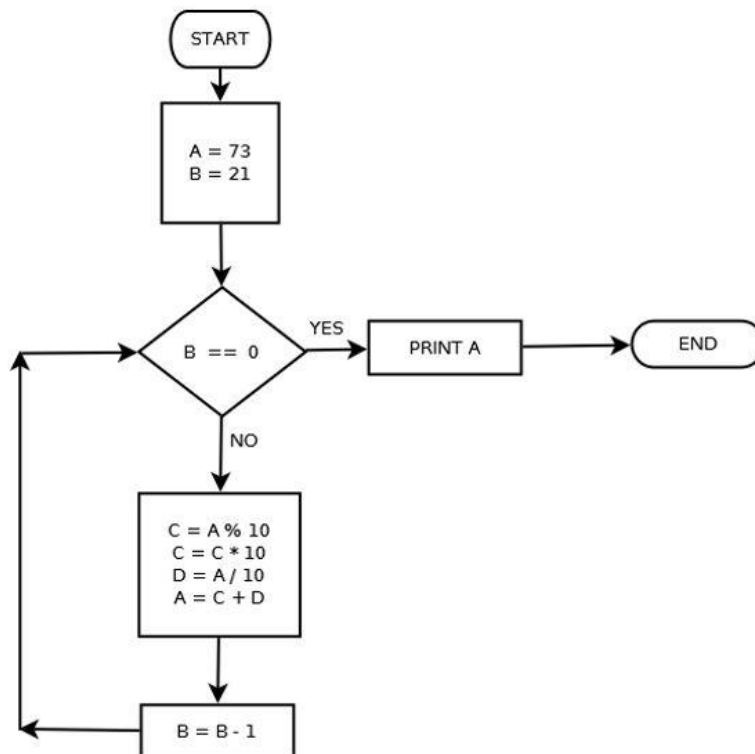
PRINT [NEWLINE] inserts a new line to the result.

PRINT [SPACE] prints a blank space in the result.

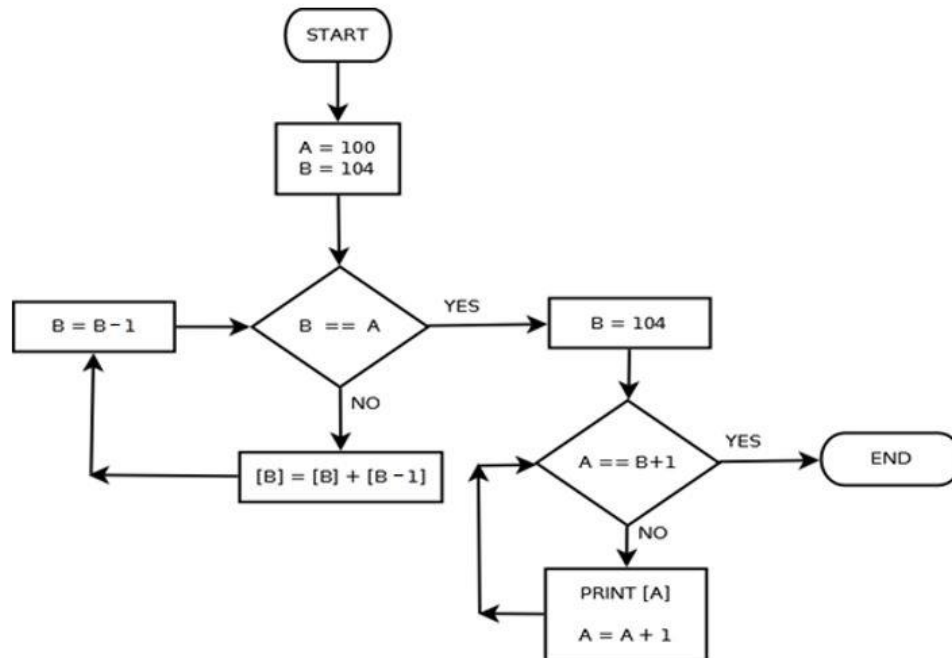
21. Find the result of the flowchart below.



22. Find the result of the flowchart below.

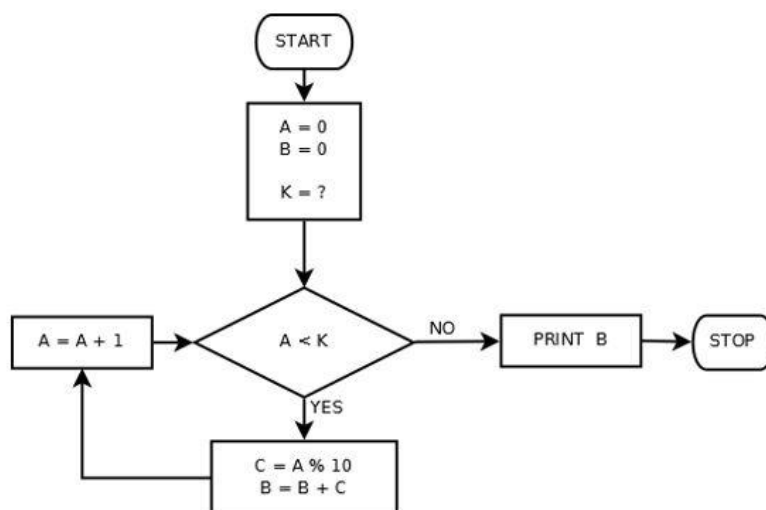


23. Find the result of the flowchart below.

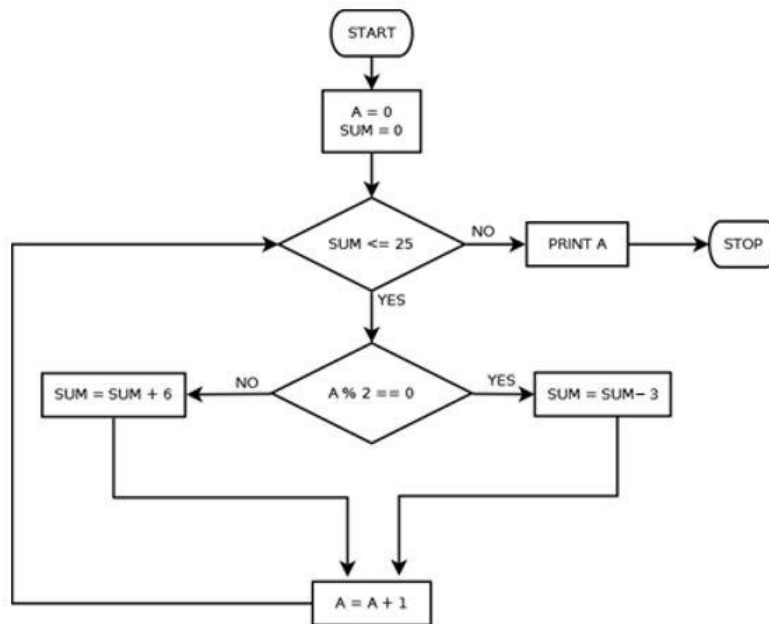


Address:	100	101	102	103	104
Value:	1	2	5	4	3

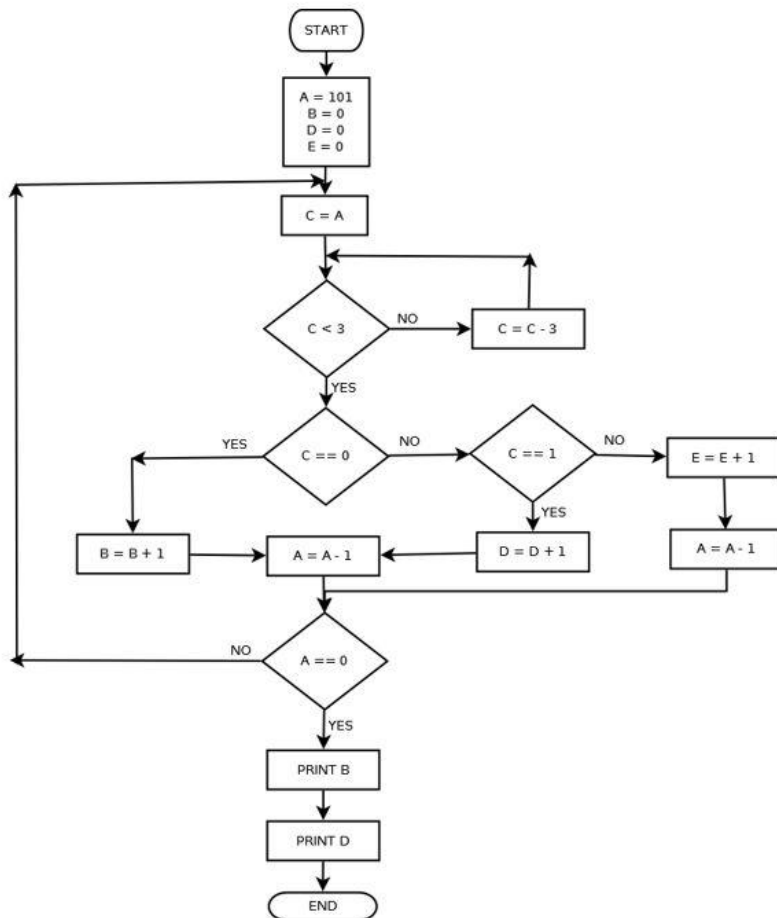
24. What must be the value of 'K' to get output as '100'.



25. Find the output of the below flowchart.

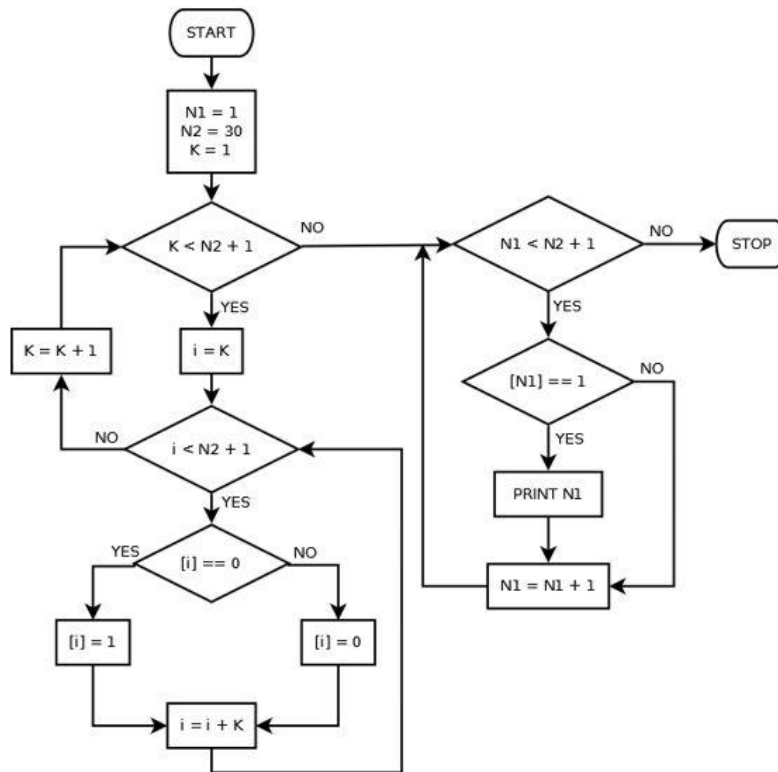


26. Find the output of the below flowchart.

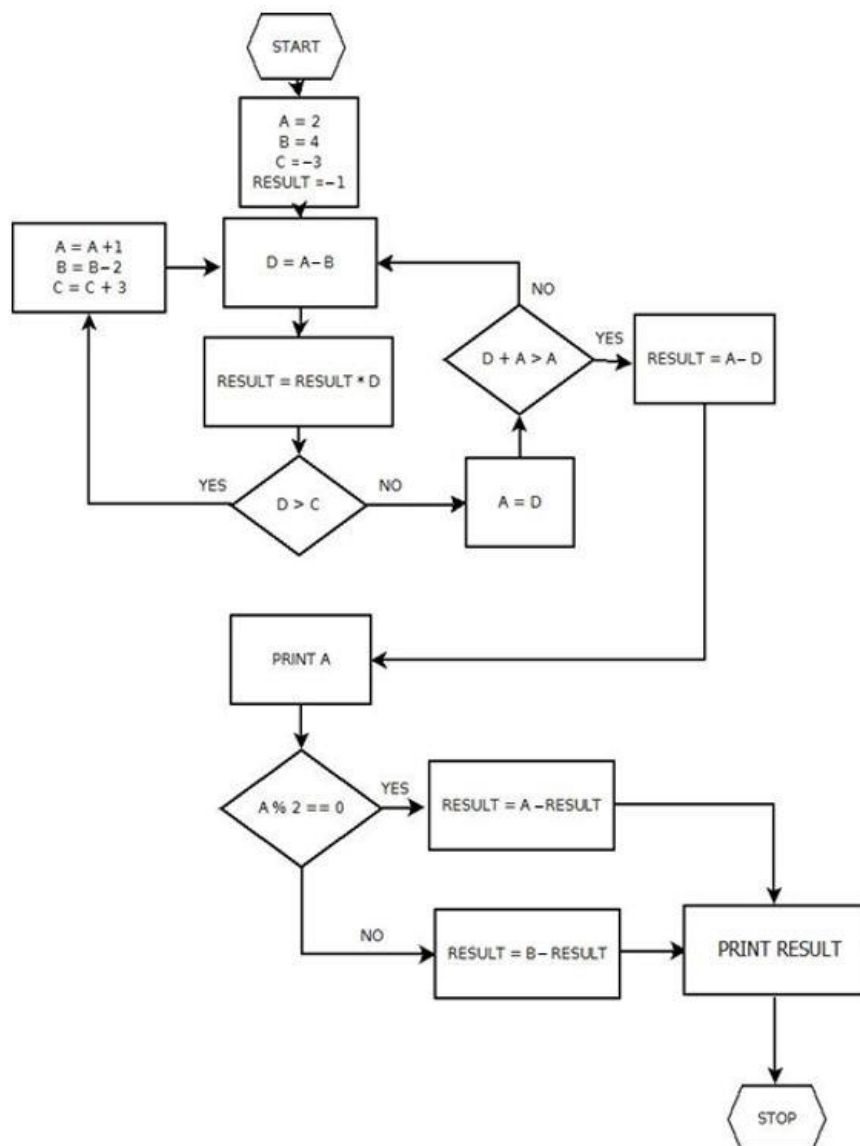


27. Find the output of the below flowchart.

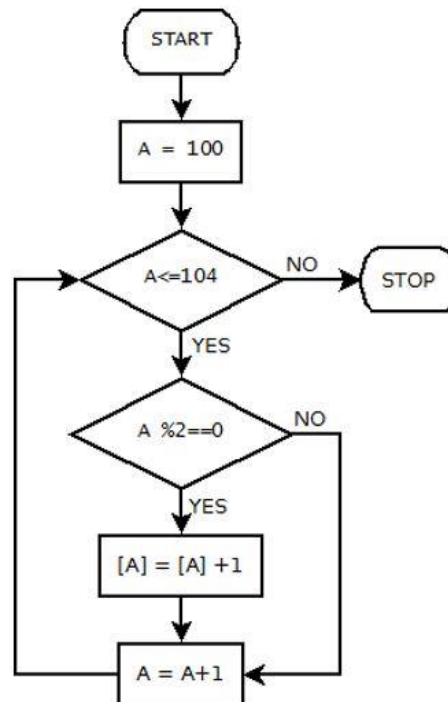
Note: All the memory locations from [1] to [30] will have 0 by default.



28. Find the output of the below flowchart.



29. What will be the value at address '103' after the below flowchart is executed?



Address	100	101	102	103	104
Value	3	8	7	6	4

30. What should be the value of 'k' to get the desired output?

Desired Output: 60

