

## Logic Building Assignment : 10

Create separate visual Studio project for each problem statement separately.

### 1. Accept N numbers from user and return frequency of even numbers.

Input : N : 6

Elements : 85 66 3 80 93 88

Output : 3

Program Layout :

```
#include<stdio.h>

int CountEven(int Arr[], int iLength)
{
    // Logic
}

int main()
{
    int iSize = 0,iRet = 0,iCnt = 0;
    int *p = NULL;

    printf("Enter number of elements");
    scanf("%d",&iSize);

    p = (int *)malloc(iSize * sizeof(int));

    if(p == NULL)
    {
        printf("Unable to allocate memory");
        return -1;
    }

    printf("Enter %d elements ",iLength);

    for(iCnt = 0;i<iLength; iCnt++)
    {
        printf("Enter element : %d",iCnt+1);
        scanf("%d",&p[iCnt]);
    }

    iRet = CountEven(p, iSize);
```

```
printf("Result is %d",iRet);

free(p);

return 0;
}
```

## 2. Accept N numbers from user and return difference between frequency of even number and odd numbers.

Input :     N :         7

             Elements : 85    66    3    80    93    88    90

Output :    1 (4 -3)

Program Layout :

```
#include<stdio.h>

Int Frequency(int Arr[], int iLength)
{
    // Logic
}

int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iRet = 0;
    int *p = NULL;

    printf("Enter number of elements");
    scanf("%d",&iSize);

    p = (int *)malloc(iSize * sizeof(int));

    if(p == NULL)
    {
        printf("Unable to allocate memory");
        return -1;
    }

    printf("Enter %d elements ",iLength);

    for(iCnt = 0;iCnt<iLength; iCnt++)
    {
        printf("Enter element : %d",iCnt+1);
```

```

        scanf("%d",&p[iCnt]);
    }

    iRet = Frequency(p, iSize);

    printf("%d",iRet);

    free(p);

    return 0;
}

```

### 3. Accept N numbers from user check whether that numbers contains 11 in it or not.

Input :     N :           6

         Elements : 85   66   11   80   93   88

Output :   11 is present

Input :     N :           6

         Elements : 85   66   3   80   93   88

Output :   11 is absent

Program Layout :

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

```
#define TRUE 1
```

```
#define FALSE 0
```

```
typedef int BOOL;
```

```
BOOL Check(int Arr[], int iLength)
```

```
{
    // Logic
}
```

```
int main()
```

```
{
    int iSize = 0,iRet = 0,iCnt = 0;
    int *p = NULL;
    BOOL bRet = FALSE;
```

```

printf("Enter number of elements");
scanf("%d",&iSize);

p = (int *)malloc(iSize * sizeof(int));

if(p == NULL)
{
    printf("Unable to allocate memory");
    return -1;
}

printf("Enter %d elements ",iLength);

for(iCnt = 0;iCnt<iLength; iCnt++)
{
    printf("Enter element : %d",iCnt+1);
    scanf("%d",&p[iCnt]);
}

bRet = Check(p, iSize);

if(bRet == TRUE)
{
    printf("11 is present");
}
else
{
    printf("11 is absent");
}

free(p);

return 0;
}

```

#### 4. Accept N numbers from user and return frequency of 11 form it.

Input :     N :           6

          Elements : 85   66   3    15   93   88

Output :   0

Input :     N :           6

          Elements : 85   11   3    15   11   111

Output :   2

Program Layout :

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

```
int Frequency(int Arr[], int iLength)
{
    // Logic
}
```

```
int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iRet = 0;
    int *p = NULL;

    printf("Enter number of elements");
    scanf("%d",&iSize);

    p = (int *)malloc(iSize * sizeof(int));

    if(p == NULL)
    {
        printf("Unable to allocate memory");
        return -1;
    }

    printf("Enter %d elements ",iLength);

    for(iCnt = 0;iCnt<iLength; iCnt++)
    {
        printf("Enter element : %d",iCnt+1);
        scanf("%d",&p[iCnt]);
    }

    iRet = Frequency(p, iSize);

    printf("%d",iRet);

    free(p);

    return 0;
}
```

**5. Accept N numbers from user and accept one another number as NO , return frequency of NO form it.**

Input : N : 6  
NO: 66  
Elements : 85 66 3 66 93 88

Output : 2

Input : N : 6  
NO: 12  
Elements : 85 11 3 15 11 111

Output : 0

Program Layout :

```
#include<stdio.h>

int Frequency(int Arr[], int iLength, int iNo)
{
    // Logic
}

int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iRet = 0, iValue = 0;
    int *p = NULL;

    printf("Enter number of elements");
    scanf("%d",&iSize);

    printf("Enter the number");
    scanf("%d",&iValue);

    p = (int *)malloc(iSize * sizeof(int));

    if(p == NULL)
    {
        printf("Unable to allocate memory");
        return -1;
    }

    printf("Enter %d elements ",iLength);

    for(iCnt = 0;iCnt<iLength; iCnt++)
```

```
{  
    printf("Enter element : %d",iCnt+1);  
    scanf("%d",&p[iCnt]);  
}  
  
iRet = Frequency(p, iSize,iValue);  
  
printf("%d",iRet);  
  
free(p);  
  
return 0;  
}
```

