

Logic Building Assignment : 23

Create separate visual Studio project for each problem statement separately.

1. Accept N numbers from user and accept one another number as NO , check whether NO is present or not.

Input : N : 6

NO: 66

Elements : 85 66 3 66 93 88

Output : TRUE

Input : N : 6

NO: 12

Elements : 85 11 3 15 11 111

Output : FALSE

Program Layout :

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

```
#define TRUE 1
```

```
#define FALSE 0
```

```
typedef int BOOL;
```

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

```
{  
    // Logic  
}
```

```
int main()
```

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

```
    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]);
}

bRet = Check(p, iSize,iValue);

if(bRet == TRUE)
{
    printf("Number is present");
}
else
{
    printf("Number is not present");
}

free(p);

return 0;
}
```

2. Accept N numbers from user and accept one another number as NO , return index of first occurrence of that NO.

Input : N : 6

 NO: 66

 Elements : 85 66 3 66 93 88

Output : 1

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

Output : -1

Program Layout :

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

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

```
int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iValue = 0,iRet = 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 = FirstOcc(p, iSize,iValue);
```

```

if(iRet == -1)
{
    printf("There is no such number");
}
else
{
    printf("First occurrence of number is %d",iRet);
}

free(p);

return 0;
}

```

3. Accept N numbers from user and accept one another number as NO , return index of last occurrence of that NO.

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

Input : N : 6
 NO: 93
 Elements : 85 66 3 66 93 88
 Output : 4

Input : N : 6
 NO: 12
 Elements : 85 11 3 15 11 111
 Output : -1

Program Layout :

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

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

```
int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iValue = 0,iRet = 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 = LastOcc(p, iSize,iValue);

    if(iRet == -1)
    {
        printf("There is no such number");
    }
    else
    {
        printf("Last occurrence of number is %d",iRet);
    }
}
```

```

    free(p);

    return 0;
}

```

4. Accept N numbers from user and accept Range, Display all elements from that range

Input : N : 6

 Start: 60

 End : 90

 Elements : 85 66 3 76 93 88

Output : 66 76 88

Input : N : 6

 Start: 30

 End : 50

 Elements : 85 66 3 76 93 88

Output :

Program Layout :

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

```

void Range(int Arr[], int iLength, int iStart, int iEnd)
{
    // Logic
}

```

```

int main()
{
    int iSize = 0,iRet = 0,iCnt = 0, iValue1 = 0,iValue1 = 0;
    int *p = NULL;

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

```

```
printf("Enter the starting point");
scanf("%d",&iValue1);

printf("Enter the ending point");
scanf("%d",&iValue2);

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 = Range(p, iSize,iValue1, iValue2);

free(p);

return 0;
}
```

5. Accept N numbers from user and return product of all odd elements.

Input : N : 6
 Elements : 15 66 3 70 10 88

Output : 45

Input : N : 6
 Elements : 44 66 72 70 10 88

Output : 0

Program Layout :

```
#include<stdio.h>
int Product(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 = Product(p, iSize);

    printf("Product is %d",iRet);

    free(p);

    return 0;
}
```