Logic Building Assignment : 17

Create separate visual Studio project for each problem statement separately.

1. Accept N numbers from user and return difference between summation of even elements and summation of odd elements.

Input : N : 6

Elements : 85 66 3 80 93 88

Output : 53 (234 - 181)

Program Layout :

```
#include<stdio.h>

int Difference(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 = Difference(p, iSize);

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

        free(p);

        return 0;
}
```

2. Accept N numbers from user and display all such elements which are divisible by 5.

Input : N : 6

Elements : 85 66 3 80 93 88

Output : 85 80

Program Layout :

```
#include<stdio.h>

void Display(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;iCnt<iLength; iCnt++)
        {
```

```
        printf("Enter element : %d",iCnt+1);
        scanf("%d",&p[iCnt]);
    }

    Display(p, iSize);

    free(p);

    return 0;
}
```

3. Accept N numbers from user and display all such elements which are even and divisible by 5.

Input : N : 6

Elements : 85 66 3 80 93 88

Output : 80

Program Layout :

```
#include<stdio.h>

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

        Display(p, iSize);

        free(p);

        return 0;
}
```

4. Accept N numbers from user and display all such elements which are divisible by 3 and 5.

Input : N : 6

Elements : 85 66 3 15 93 88

Output : 15

Program Layout :

```
#include<stdio.h>

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

        Display(p, iSize);

        free(p);

        return 0;
}
```

5. Accept N numbers from user and display all such elements which are multiples of 11.

Input : N : 6

Elements : 85 66 3 55 93 88

Output : 66 55 88

Program Layout :

```
#include<stdio.h>

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

        Display(p, iSize);

        free(p);

        return 0;
}
```