Logic Building Assignment : 16

Create separate visual Studio project for each problem statement separately.

1. Accept N numbers from user and return the largest number.

Input : N : 6

Elements : 85 66 3 66 93 88

Output : 93

Program Layout :

```
#include<stdio.h>

#define TRUE 1
#define FALSE 0

typedef int BOOL;

int Maximum(int Arr[], int iLength)
{
	// 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 = Maximum(p, iSize);

        printf("Largest Number is %d",iRet);

        free(p);

        return 0;
}
```

2. Accept N numbers from user and return the smallest number.

Input : N : 6

Elements : 85 66 3 66 93 88

Output : 3

Program Layout :

```
#include<stdio.h>

#define TRUE 1
#define FALSE 0

typedef int BOOL;

int Minimum(int Arr[], int iLength)
{
        // 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 = Minimum(p, iSize);

        printf("Smallest Number is %d",iRet);

        free(p);

        return 0;
}
```

3. Accept N numbers from user and return the difference between largest and smallest number.

Input : N : 6

Elements : 85 66 3 66 93 88

Output : 90 (93 -3)

Program Layout :

```
#include<stdio.h>

#define TRUE 1
#define FALSE 0

typedef int BOOL;
```

```
int Difference(int Arr[], int iLength)
{
	// 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 = Difference(p, iSize);

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

	free(p);

	return 0;
}
```

4. Accept N numbers from user and display all such numbers which contains 3 digits in it.

Input : N : 6

Elements : 8225 665 3 76 953 858

Output : 665 953 858

Program Layout :

```
#include<stdio.h>


void Digits(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]);
	}

	Digits(p, iSize);

	free(p);

	return 0;
}
```

5. Accept N numbers from user and display summation of digits of each number.

Input : N : 6

Elements : 8225 665 3 76 953 858

Output : 17 17 3 13 17 21

Program Layout :

```
#include<stdio.h>

void DigitsSum(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]);
	}

	DigitsSum(p, iSize);

	free(p);

	return 0;


}
```