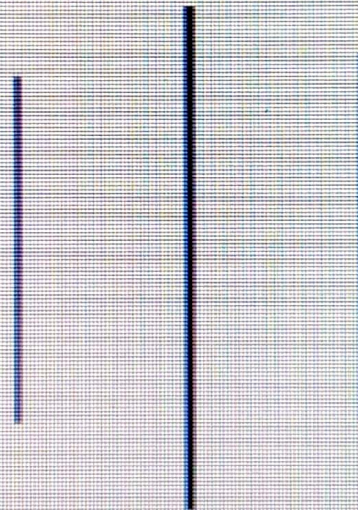


**INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING
& MANAGEMENT
KUPONDOLE, LATIPUR
(AFFILIATED TO TRIBHUVAN UNIVERSITY)**



LAB REPORT

LAB NO.: 8

SUBJECT: C PROGRAMMING

SUBMITTED BY:

NAME: ASHWANI KUMAR CHAUDHARY
ROLL NO.: 019
DATE: 2078/01/01

SUBMITTED TO:

**DEPARTMENT OF
COMPUTER &
ELECTRONICS**


```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a[100], n, i, s=0;
    printf("How many no. do you want? \n");
    scanf("%d", &n);
    printf("Enter the elements \n");
    for (i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    for (i=0; i<n; i++)
    {
        s = s + a[i];
    }
    printf("sum is %d", s);
    getch()
}

```

Output

How many no. do you want?

3

Enter the elements

34

32

54

Sum is 120

```
#include <stdio.h>
#include <conio.h>
void read (int T[], int);
void sort (int T[], int);
void display (int T[], int);
void main ()
{
    int a[100], i, j, n, num;
    printf("How many no.? \n");
    scanf ("%d", &n);
    read (a, n);
    sort (a, n);
    printf("Element after sorting is \n");
    display (a, n);
}
```

```
void read (int a[100], int n).
```

```
{
    int i;
    printf("Number are \n");
    for (i=0; i<n; i++)
    {
        scanf ("%d", (a+i));
    }
}
```

```
void sort (int a[100], int n)
```

```
{
    int i, j, temp;
    for (i=0; i<n-1; i++)
    {
        for (j=0; j<n-i-1; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}
```

```
temp = * (a+j);  
*(a+j) = *(a+j+1);  
*(a+j+1) = temp;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void display (int a[100], int n)
```

```
{
```

```
int i;
```

```
for (i=0; i<n; i++)
```

```
{
```

```
printf ("%d\n", *(a+i));
```

```
}
```

```
}
```

~~Output~~

#include <stdio.h>

#include <conio.h>

void find (int, int, int, int*);

void main ()

{

int a, b, c, max, min;

printf ("Enter three no.");

scanf ("%d %d %d", &a, &b, &c);

find (a, b, c, &max);

printf ("Max is %d", max);

}

void find (int a, int b, int c, int*)

{

if (a > b && a > c)

{

*x = a;

}

else if (b > a && b > c)

{

*x = b;

}

else

{

*x = c;

}

}

Output

Enter three no.

23

12

21

Max is 23.

```

#include <stdio.h>
#include <conio.h>
void reverse_array (int a[], int n);
void main()
{
    int a[100], n, i, temp, j;
    printf("How many no. you want-? \n");
    scanf("%d", &n);
    for (i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }

    printf("Array is: \n");
    for (i=0; i<n; i++)
    {
        printf("%d \n", a[i]);
    }

    reverse_array(a, n);
    printf("\n array in reverse order is: \n");
    for (i=0; i<n; i++)
    {
        printf("%d \n", a[i]);
    }
    getch();
}

void reverse_array (int a[100], int n)
{
    int i, temp;
    for (i=0; i<(n/2); i++)
    {

```

```
temp = *(a+i);  
*(a+i) = *(a+n-i-1);  
*(a+n-i-1) = temp;  
}
```

}

Out put

How many no. you want? 4

4

5

8

9

Array is

4

5

8

9

Array in reverse order is:

9

8

5

4.

```

#include <stdio.h>
#include <conio.h>
void swap (int*, int*);
void main()
{
    int a, b;
    int *pa, *pb;
    pa = &a, pb = &b;
    printf("Enter the no. ");
    scanf("%d %d", pa, pb);
    printf("The no. before swapping is:\n");
    printf("a = %d & b = %d", *pa, *pb);
    swap(pa, pb);
    printf("\n The no. after swapping is:\n");
    printf("a = %d & b = %d", *pa, *pb);
    getch();
}

void swap (int *a, int *b)
{
    int int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

```

Output

```

Enter the no: 3, 1

The no. before swapping is:
a = 3 & b = 1

The no. after swapping is:
a = 1 & b = 3

```


Discussion & Conclusions.

With the theoretical knowledge of string. We were able to perform various program with the help of ~~various~~ pointer in c-programming. We were able to pass words to a program. & can manipulate the given word & even found out the no. of character. Similarly in this lab we were also able to ask some random word & sort them dictio order. Thus we were able to use pointers in c-programming