JEPPIAAR INSTITUTE OF TECHNOLOGY

"Self-Belief | Self Discipline | Self Respect"

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

LECTURE NOTES

CS8251 – C PROGRAMMING

(Regulation 2017)

Year/Semester: I/II CSE

2020 – 2021

Prepared by

Mr.H.Shine

Assistant Professor/CSE

UNIT-II - ARRAYS AND STRINGS:-

Introduction to Arrays: Declaration, Initialization One dimensional array - Example program: Computing Mean, median, and mode - Two dimensional arrays - Example program: Matrix operations (Addition, scaling, Determinant and Transpose) - String operations: length, Compare. Concatenate, Copy - Selection sort, linear and binary search.

Introduction to Arrays:-

✓ An array is a group (or collection) of same data types elements stored under common name.

✓ For example an int array holds the elements of int types while a float array holds the elements of float types.

Declaration:-

Syntax:-

```
datatype arrayName[arraysize];
```

✓ for example

```
int mark[5];
```

✓ Here we declared an array, mark of

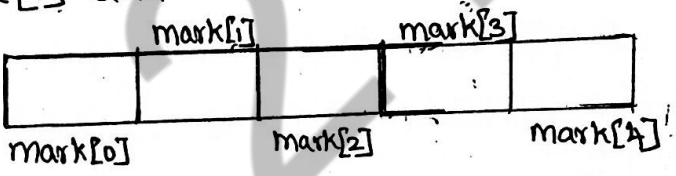
integer datatype. And its size is 5. Meaning it can hold 5 integer values.

✓ It's important to note that the size and type of an array cannot be changed once it is declared.

Access Array Elements:-

✓ you can access elements of an array by indices or array subscript.

✓ suppose you declared an array mark as above. The first element is mark[0], the 2nd element is mark[1] and so on.



✓ Arrays have 0 as the first index, not 1. In this example, mark[0] is the first element.

✓ If the size of an array is n, to access the last element, the n-1 index is used. In this example, mark[4].

✓ Suppose the starting address of mark[0] is 2120. Then the address of the mark[1] will be 2122. Similarly, the address of mark[2] will be 2124 and so on. This is because the size of a integer is 2 bytes.

Initialization:-

✓ A simple way to initialize array is by index.

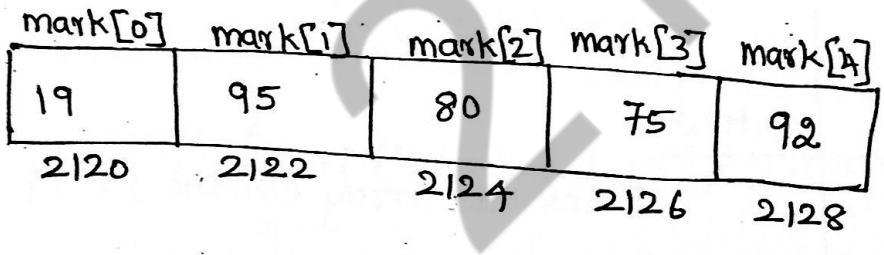
✓ It is possible to initialize an array during declaration. For example.

```
int mark[5] = {19,95,80,75,92};
```

✓ you can initialize an array like this

```
int mark[] = {19,95,80,75,92};
```

✓ Here, we haven't specified the size. However the Compiler knows its size is 5. as we are initializing it with 5 elements.



Input and output Array Elements:-

→ Input data into array:-

✓ Here we are using iteration statements for read the array elements by using scanf() function.

✓ For example we are iterating the array from 0 to 4 because the size of the array is 5. Inside the loop we are display-ing a message to the user to enter the values.

✓ All the input values are stored in the Corresponding array elements using scanf function.

Example program:-

```
#include <stdio.h>
int main()
 {
int mark[5], i;
for(i=0; i<5; i++)
  {
printf("\nEnter the array element%d:", i+1);
scanf("%d", &mark[i]);
  }
return 0;
}
```

output:-

Enter the array element 1: 19

Enter the array element 2: 95

Enter the array element 3: 80

Enter the array element 4: 75

Enter the array element 5: 92

Read out data from an array:

✓ Here we are using iteration or looping statements for read out array elements and using printf() function.

✓ For example we are iterating the array from 0 to 4. because the size of the

array is 5. Inside the loop we are displaying the marks to the user by using printf function.

Example program:-

```
#include <stdio.h>
int main()
{
int mark[5]={19,95,80,75,92};
int i;
for(i=0;i<5;i++)
{
printf("mark %d is %d", i+1, mark[i]);
}
return 0;
}
```

Output:-

```
mark 1 is 19
mark 2 is 95
mark 3 is 80
mark 4 is 75
mark 5 is 92
```

Example Program:- [Findout mean and median]

```
#include <stdio.h>
int main()
{
int i,j,n,t;
float x,y,sum=0,a[20];
printf("\n Enter the limit \n");
scanf("%d",&n);
printf("Enter the elements \n");
for(i=0;i<n;i++)
 {
  scanf("%f",&a[i]);
   sum=sum+a[i];
   }
 x=sum/n;
printf("Mean\t = %f",x);
for(i=0;i<n;i++)
 {
 for(j=i+1;j<n;j++)
  {
   if(a[i]>a[j])
     {
      t=a[i];
     a[i]=a[j];
     a[j]=t;
     }
  }
 }
```

```
if (n % 2 == 0)
    {
    y = (a[n/2] + a[(n-1)/2])/2;
    }
else
    {
    y = a[(n-1)/2];
    printf("\n Median \t = %f ", y);
    return 0;
}
```

Sample output 1:-

```
Enter the limit
6
Enter the elements
1
2
1
3
4
3
Mean = 2.333333
Median = 2.500000
```

Output 2:-

```
Enter the limit
5
Enter the elements
1
6
2
4
3
Mean = 3.200000
Median = 3.000000
```

Example Program: [Find out mode] :-

```
#include <stdio.h>
int main()
{
int i,j,n,a[20],b[20],k,c=1,max=0,mode;
printf("\n Enter the limit \n");
scanf("%d",&n);
printf("Enter the Elements \n");
for(i=0; i<n; i++)
 {
  scanf("%d",&a[i]);
 }
for(i=0; i<n-1 ; i++)
 {
 mode=0;
for (j=i+1;j<n; j++)
  {
  if(a[i]==a[j])
    {
    mode++;
    }
  }
 if((mode>max)&&(mode!=0))
  {
  k=0;
  max=mode;
  b[k]=a[i];
   k++;
  }
```

```
else if (mode == max)
    {
     b[k] = a[i];
     k++;
    }
  }
for(i=0; i<n; i++)
  {
   if(a[i] == b[i])
    c++;
  }
 if(c == n)
  {
   printf("\n There is no mode");
  }
 else
  {
   printf("\n mode \t = ");
   for(i=0; i<k; i++)
     {
      printf("%d", b[i]);
     }
  }
 return 0;
}
```

Sample output1:-

```
Enter the limit
6
Enter the elements
5
6
1
```

1
6
2

mode = 6 |

Output 2:-

Enter the limit

8

Enter the elements

3
6
4
2
1
2
1
2

Mode = 2

8M

Two Dimensional Arrays - Example programs:-

Matrix - Addition:-

```
#include<stdio.h>
int main()
{
int r, c, a[50][50], b[50][50], sum[50][50], i, j;
printf("Enter the no. of rows:\t");
scanf("%d", &r);
printf("Enter the no. of Columns:\t");
scanf("%d", &c);
printf("\n Enter elements of 1st matrix:\n");
for(i=0; i<r; i++) {
    for(j=0; j<c; j++)
    {scanf("%d", &a[i][j]); }
```

```
    }
printf("\n Enter elements of 2nd matrix :\n");
 for(i=0; i<r; i++)
    {
     for(j=0; j<c ; j++)
       {
        scanf("%d" ,&b[i][j]);
       }
     }
   for(i=0; i<r ; i++)
     {
      for(j=0; j<c ;j++)
        {
         sum[i][j]= a[i][j] + b[i][j];
        }
     printf("Sum of two matrices:\n");
     for(i=0; i<r; i++)
       {
        for(j=0; j<c; j++)
          {
           printf("%d  ", sum[i][j]);
          }
        printf("\n");
       }
     return 0;
   }
```

Sample output 1:-

Enter the no. of rows: 2

Enter the no. of columns: 2

Enter the elements of 1st matrix:

1
1
1
1

Enter the elements of 2nd matrix:

2
2
4
5

Sum of two matrices:

3 3
5 6

output 2:-

Enter the no. of rows: 3

Enter the no. of columns: 3

Enter the elements of 1st matrix:

1
2
3
1
4
2
5
6
1

Enter the elements of 2nd matrix:

2
1
4

```
5
7
3
1
2
2
```

Sum of two matrices:

```
3  3  7
6  11 5
6  8  3
```

8M

Example Program - Transpose of matrix:-

```
#include<stdio.h>
int main() {
int a[10][10], transpose[10][10], r, c, i, j;
printf("Enter rows and Columns:");
scanf("%d %d", &r, &c);
printf("\n Enter matrix elements:\n");
for(i=0; i<r ; i++)
  {
   for(j=0; j<c ; j++)
    {
     printf("Enter element a%d%d:",
                    i+1, j+1);
     scanf("%d", &a[i][j]);
    }
  }
printf("\n Entered matrix:\n");
```

```
for(i=0; i<r ; i++)
  {
   for (j=0 ; j<c ; j++)
     {
      printf ("%d ", a[i][j]);
     }
    printf("\n");
  }
for(i=0; i<r ; i++)
  {
   for (j=0; j<c ; j++)
     {
      transpose[j][i] = a[i][j];
     }
  }
printf ("\n Transpose of the matrix :\n");
for (i=0 ; i<c ; i++)
   {
    for (j=0; j<r ; j++)
      {
       printf ("%d", transpose[i][j]);
      }
    printf("\n");
   }
 return 0;
}
```

Output:-

```
Enter rows and columns : 2
3
Enter matrix elements :
Enter element a11 : 1
Enter element a12 : 2
Enter element a13 : 3
Enter element a21 : 4
Enter element a22 : 5
Enter element a23 : 8
Entered matrix :
1 2 3
4 5 6
Transpose of the matrix
1 4
2 5
3 8
```

3M

String Operations:-

String:-

✓ Array of character is called a string.

It is always terminated by the NULL character.

String is a one dimensional array of character.

✓ We can initialize the string as

```
Char name[] = {'h','e','l','l','o','\0'};
```

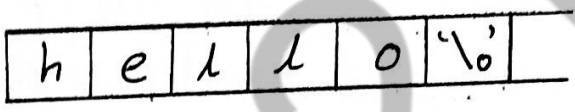
✓ Here each character occupies 1 byte of memory and last character is always NULL character.

✓ where '\0' and 0 (zero) are not same, where ASCHII value of '\0' is 0 and ASCHII value of 0 is 48.

✓ Array elements of character array are also stored in contiguous memory allocation.

✓

From the above we can represent as;



✓ string can also be initialized as

```
char name[] = "hello";
```

✓ Here null character is not necessary and the compiler will assume it automatically.

String Library functions:-

✓ There are several string library functions used to manipulate string and the prototypes for these functions are in header file "string.h". several string functions are

① strlen() :-

✓ This function return the length of the

string. (i.e) the number of characters in the string excluding the terminating NULL character.

✓ It accepts a single argument which is pointer to the first character of the string.

✓ For example

strlen ("welcome");

→ It returns the valve 7.

example Program:-

```
#include <stdio.h>
#include <string.h>
void main()
{
char str[50];
printf("Enter a string:\n");
gets(str);
printf("Length of the string is %d\n",
strlen(str));
getch();
}
```

read the string from user

Output:-

```
Enter a string
welcome
Length of the string is 7.
```

2) Strcmp( ):-

✓ This function is used to compare two strings. If two string match, strcmp() return a value 0. otherwise it return a non-zero value.

✓ It compare the strings character by character and the comparison stops when the end of the string is reached or the corresponding characters in the two string are not same.

```
strcmp(S1, S2);
```

example Program:-

```
#include <stdio.h>
#include <string.h>
void main()
  {
char str[10], str2[20];
printf("Enter two strings \n");
 gets(str1);
 gets(str2);
if(Strcmp(str1, str2)==0)
 {
 printf("String are same \n");
 }
```

```
else
  {
  printf("String are not same \n");
  }
getch();
}
```

Output:-

```
Enter two strings

hello
hello
String are same.
```

3) strcpy() :-

✓ This function is used to copying one string to another string.

✓ The function strcpy (str1, str2) copies str2 to str1 including the NULL character.

✓ Here str2 is the source string and str1 is the destination string.

✓ The old content of the destination string str1 are lost. The function returns a pointer to destination string str1.

Example Program:-

```
#include <stdio.h>
#include <string.h>
Void main()
{
Char str1[10], str2[10];
printf ("Enter a string \n");
scanf ("%s", str2);
strcpy (str1, str2);
printf ("First string : %s \t second
        String : %s \t ", str1, str2);
strcpy (str1, "Delhi");
strcpy (str2, "Bangalore");
printf (" First string : %s \t Second string %s",
                str1, str2);
getch();
}
```

Output:-

```
Enter a string
Welcome
First string : Welcome  second string :
                welcome
First string : Delhi  second string :
                Bangalore.
```

(4) Strcat():-

✓This function is used to append a copy of a string at the end of the other string.

✓If the first string is "hello" and second string is "welcome" then after using this function the string becomes "hellowelcome"

✓The NULL character from str1 is moved and str2 is added at the end of str1.

✓The 2nd string str2 remains unaffected. A pointer to the first string str1 is returned by the function.

Example Program:-

```
#include <stdio.h>
#include <string.h>
void main()
{
char str1[20], str2[20];
printf("Enter two strings\n");
gets(str1);
gets(str2);
strcat(str1, str2);
printf("First string:%s\t Second string:%s
str1, str2);
```

```
                                    ↗ both data & base
    strcat(str1, "_one");
    printf("Now string is:%s \t", str1);
}   getch();
```

Output:-

```
Enter two strings
data
base
First string : database
Now first string database_one
```

8M

Write a 'C' program to perform the Selection Sort? :-

```
#include <stdio.h>
void main()
{
int a[100], n, i, j, t;
printf("Enter no. of elements\n");
scanf("%d", &n);
printf("Enter the elements are \n");
for(i=0; i<n; i++)
 {
 scanf("%d", &a[i]);
}}
for(i=0; i<n-1; i++)
  {
```

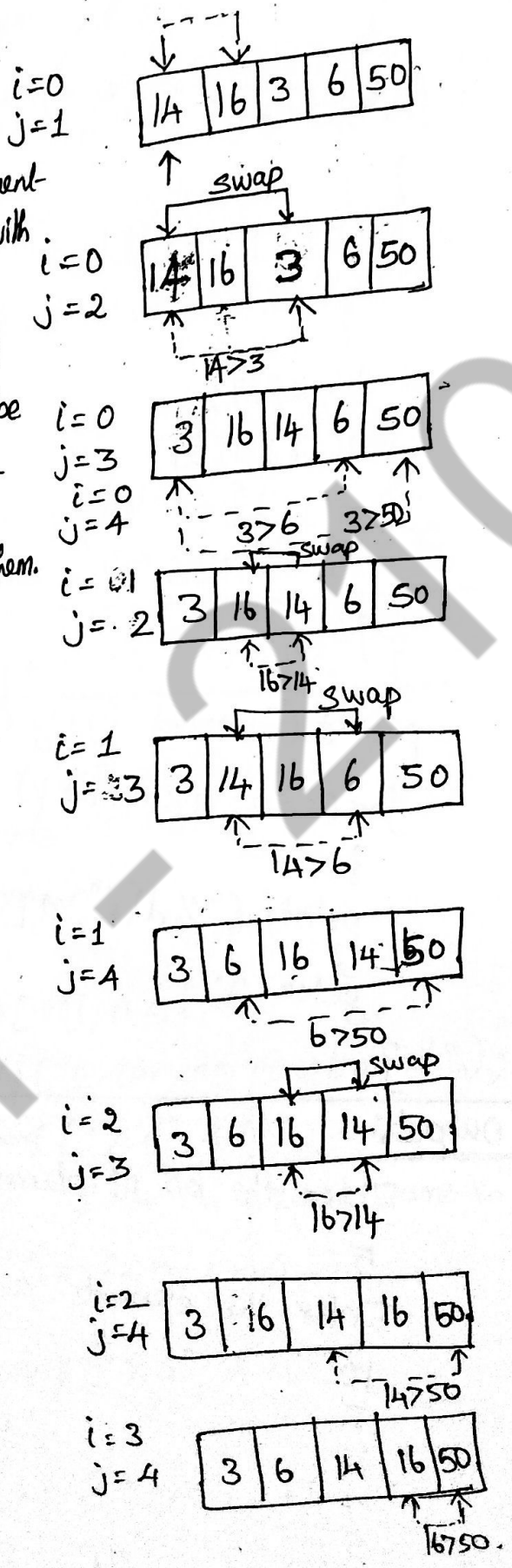
```
for(j=i+1; j<n; j++)
 {
 if(a[i] >a[j])
   {
   t=a[i];
   a[i] =a[j];
   a[j] =t;
   }
 }
}
printf("Sorted Array elements \n");
 for(i=0; i<n; i++)
   {
   printf("%d\t", a[i]);
   }
   getch();
 }
```

Output:-

```
Enter the no. of elements
5
Enter the elements are
10
2
18
7
5
Sorted Array elements
2  5  7  10  18
```

* Take the 1st element and compare it with other element.

→ If you find any element to be smaller than the 1st element, then swap both of them.



Write a 'c' program to perform the linear search:-

```
#include <stdio.h>
void main()
  {
  int a[20], n, i, key, found=0;
  printf("Enter the no. of elements \n");
  scanf("%d\t", &n);
  printf("Enter the elements \n");
  for(i=0; i<n; i++)
    {
    scanf("%d", &a[i]);
  }
  printf("Enter the search element \n");
  scanf("%d", &key);
  for(i=0; i<n; i++)
    {
    if(a[i]==key)
      {
      found=1;
      }
    }
  if(found==1)
    {
  printf("element found in list \n");
```

```
    }
else
  {
   printf("element not found in list\n");
  }
getch();
}
```

output:-

```
Enter the no. of elements
6
Enter the elements
15
12
8
7
5
25
Enter the search element
7
Element found in the list.
```

Write a 'c' program to perform Binary Search:-

```
#include<stdio.h>
void main()
{
int a[25], n, i, key, t, j, found=0, low, mid, high;
printf("Enter the no. of elements\n");
scanf("%d\t", &n);
```

```
printf("Enter the elements \n");
for(i=0; i<n; i++)
 {
 scanf("%d", &a[i]);
 }
printf("Enter the key search element \n");
 scanf("%d \t", &key);

for (i=0; i<n-1; i++)
 {
 for (j=i+1; j<n; j++)
  {
  if (a[i]>a[j])
    {
      t=a[i];
      a[i]=a[j];
      a[j]=t;
      }
    }
 }
printf("Sorted elements are \n");
for (i=0; i<n; i++)
  {
 printf("%d\n", a[i]);
  }
```

```
low = 0;
high = n-1;
while (low <= high)
    {
    mid = (low + high)/2;
      if (key == a[mid]);
          {
          found = 1;
          break;
          }
        else if (key > a[mid])
          {
          low = mid+1;
          }
        else if (key< a[mid])
          {
          low = mid-1;
          }
      }
if(found == 1)
  {
  printf("Element is found");
  }
else
  {
  printf ("Element is not found");
  }
```

```
getch();
}
```

output:-

```
Enter the no. of elements
6
Enter the elements
12
20
5
25
18
Enter the key search element
25
Sorted elements are
5
12
18
20
25
Element is found.
```