Practical No. 7

PAGE NO. 39
DATE : 5/11/24

Aim:- Program to understand sorting and searching using array.

Software Required:- Turbo C/C++

Theory:-

Sorting:- Sorting is a process of ordering individual element of a list according to their proper rank, either sort a list in ascending or descending order. We can properly, easily sort a list of elements by means of iteration / loop and if-condition check statements. Sorting algorithms can be implemented by any programming languages.

Given an array of integers, print the array in such a way that the first element is first maximum and second element is first minimum & so on.

Examples:-

Input : arr[] = {7, 1, 2, 3, 4, 5, 6}
Output : 7 1 6 2 5 3 4

Searching:-

Input : arr{1, 6, 9, 4, 3, 7, 8, 2}
Output : 9 1 8 2 7 3 6 4

Instructer's Sign

- **Output of the Code 1**

```
Enter the number of elements: 5
Enter the elements : 2
4
6
1
7
Smallest element = 1
```

Practical No.

PAGE NO. 40
DATE :

Code :-

1.
```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
int x[10], i, n, min;
printf("Enter the number of elements:");
scanf("%d", &n);
printf("Enter the elements : ");
for(i=0; i<n; i++)
{
scanf("%d", &x[i]);
}
min = x[0];
for(i=1; i<n; i++)
{
if(min > x[i])
{
min = x[i];
}
}
printf("Smallest element = %d", min);
getch();
}
```

Output of the Code 2

Enter the number of element : 5
Enter the element: 2
4
6
1
7

Largest element = 7

Practical No.

PAGE NO. 41
DATE :

2.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
int x[10], i, n, max;
printf("Enter the number of elements:");
scanf("%d", &n);
printf("Enter the elements :");
for(i=0; i<n; i++)
{
scanf("%d", &x[i]);
}
max = x[0];
for(i=1; i<n; i++)
{
if(max < x[i])
{
max = x[i];
}
}
printf("Largest element = %d", max);
getch();
}
```

Output of the code 3

```
Enter the number of element: 5
Enter the element: 2
3
9
4
1
Array in Ascending order:
1 2 3 4 9
```

Practical No.

PAGE NO. 41
DATE:

3.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
int x[10],i,j,n,t;
printf("Enter the number of elements:");
scanf("%d",&n);
printf("Enter the elements:");
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++);
{
if(x[i]>x[j])
{
t=x[i];
x[i]=x[j];
x[j]=t;
}
}
printf("Array in Ascending order:\n");
for(i=0;i<n;i++)
{
printf("%d\t",x[i]);
}
getch();
}
```

Output of the Code 4

```
Enter the number of element : 5
Enter the element: 2
3
9
4
1
Array in Descending order:
1  2  3  4  9
```

Practical No.

PAGE NO. 43.
DATE :

4.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
int x[10], i, j, n, t;
printf("Enter the number of element:");
scanf("%d", &n);
printf("Enter the element:\n");
for(i=0; i<n; i++)
  {
  scanf("%d", &x[i]);
  }
for(i=0; i<n; i++)
  {
  for(j=i+1; j<n; j++)
    {
    if(x[i] < x[j])
      {
      t = x[i];
      x[i] = x[j];
      x[j] = t;
      }
    }
  }
printf("Array in Descending Order:\n");
for(i=0; i<n; i++)
  {
  printf("%d\t", x[i]);
  }
getch();
}
```

Instructer's Sign

Practical No. [ ]

PAGE NO. 44
DATE :

Viva Question:-

1. If we are having an int array and the starting address is 2000 what will be the next address of second element?

Ans If address is starting with 2000 then next address should be 2004 for an int data type
* It depends on the operating system. If it is 32 bit, it will take 4 bytes and if it is 64 bit, it will take 4 bytes.

2 We are having What are the advantages of an array?

Ans Array have many advantages, including:

- Efficient access- Array offers constant-time access to elements.
- Memory Efficiency - Arrays allocate memory in a contiguous block, which can improve cache performance
- Ordered collection - Arrays maintain the order of elements
- Simplicity - Arrays are straight forward to use & implement in most programming languages.
- Represent matrics - Array can be used to represent multiple data time of the same type using a single name.
- Efficient data storage & retrieval:- Arrays offer fast and direct access to elements using their index.

Instructer's Sign

Q3. What are the disadvantage of an array in C programming language?

Ans Array have some disadvantage -

- Fixed size - Once an array is created, its size cannot be changed. If you need more space you must create a new array & copy the element over.
- Memory waste - If you allocate an array that is larger than needed, you waste memory space. If you understimate the amount of space needed. You might run out of room.
- Inefficient insertions & deletions - Inserting & deleting elements especially in middle of an array, can be slow.
- Single data type only - Arrays are designed to store elements of the same data type
- Limited functionality for complex problems - Array may not be ideal for complex problem.

Q4 Can we change the size of an array at runtime?

Ans No, in most programming language, the size of an array cannot be changed directly during runtime.

Q5 Can we declare array size as a negative number?

No, we cannot declare an array size as a negative number in C language

Instructer's Sign

Output of the Code 1

```
Enter two numbers: 6
9
Product = 54
```

Practical No. 8

PAGE NO. 46
DATE: 12/11/24

Aim:- To learn functions and recursive functions

Software required:- Turbo C/C++

Theory:- Recursion is a programming technique that allows the programmer to express operations in terms of themselves. In C, this takes the form of a function that calls itself. A useful way to think of recursive functions is to imagine them as a process being performed where one of the instructions is to "repeat the process".

Code:-

1.
```
#include<stdio.h>
#include<conio.h>
void mul();
void main()
{
 clrscr();
 mul();
 getch();
}
void mul()
{
 int a,b,c;
 printf("Enter two numbers:");
 scanf("%d %d", &a,&b);
 c = a*b;
 printf("Product = %d",c);
}
```

Instructer's Sign

Output of code 2

Enter two numbers : 6
9
product = 54

Practical No.

PAGE NO. 47.
DATE :

2.
```
#include<stdio.h>
#include<conio.h>
void mul(int,int);
void main()
{
clrscr();
int a,b;
printf("Enter two numbers:");
scanf("%d %d",&a,&b);
mul(a,b);
getch();
}
void mul(int x,int y)
{ int c;
  c=x*y;
  printf("Product = %d", c);
}
```

Output of code 3

Enter a number: 4

Factorial = 24

Practical No.

PAGE NO. 48
DATE :

3.
```
#include <stdio.h>
#include <conio.h>
long factor();
void main()
{
f = factor();
printf(" Factorial = %ld", f);
getch();
}

long factor()
{
int n, i, z=1;
printf(" Enter a number: ");
scanf("%d", &n);
for(i=1; i<=n; i++)
    {
    z = z*i;
    }
return z;
}
```

Output of the code 4

Enter a number: 4

factorial = 24

Practical No.

PAGE NO. 49.
DATE :

4.
```
#include<stdio.h>
#include<conio.h>
long factor(int);
void main()
{
int n;
printf("Enter a number: ");
scanf("%d",&n);
f = factor(n);
printf("Factorial of %d = %ld",n,f);
getch();
}

long factor(int x)
{
int i,z=1;
for(i=1;i<x;i++)
{
z=z*i;
}
return z;
}
```

Output of the Code 5:-

Enter a number = 4

Factorial = 24

Practical No.

PAGE NO. 50.
DATE :

5.
```
#include<stdio.h>
#include<conio.h>
long factor(int);
void main()
{                                   // recursive function.
int n;
printf("Enter a number:");
scanf("%d",&n);
f = factor(n);
printf("Factorial of %d= %ld", n,f);
getch();
}
long factor(int x)
{
if (x==0' || x==1)
    {
    return 1;
    }
else
    {
    return x * factor(x-1);
    }
}
```

Viva Question :-

Q1 What do you mean by function & what are their advantages?

Ans A function in computer programming is a set of state-ments that performs a task by taking inputs, performing operations and producing results. It is also known as procedure, method, subroutine, routine, subprogram.
It provides a powerful mechanism for organizing & structure code. Functions allow developers to break down complex task into smaller, manageable components, promoting code reusability and maintainability.

Q2 How many type of functions are there in "C" programming language?

Ans The function is of two types - user defined & library function. In function we can according to two types call by value & call by reference according to the values passed.

Q3 What do you mean by recursive function?

Ans A recursive function is a function that uses its own previous term to calculate subsequent terms, forming a sequence. It is called by itself.

Instructer's Sign

Q4 What are the difference between recursion & iteration?

Ans Recursion is a technique in which the function calls itself in its body to solve the program, typically breaking into smaller & more manageable sub-problems. In contrast, iteration is a technique that repetitively executes a code block until the condition is unmet.

Q5 What is the difference between "call by value" and "call by reference" methods of function calling?

Ans The main difference between "call by value" and "call by reference" is how data is passed to a function.

Call by value :- A copy of the data is passed to the function, so changes made within the function do not affect modify the original data.

Call by reference :- The memory address (reference) of the original data is passed, so functions can directly modify the original data.

Instructer's Sign