

P

R

A

C

T

I F

C I

A L

L E



UPES

UNIVERSITY OF TOMORROW

**Submitted
by:**

RISHABH
SHARMA
500102032
B.TECH CSE
AIML-HONOUR'S

**Submitted
to:**

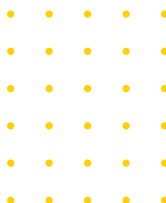
ASSISTANT
PROFESSOR(SOCS)
VIRENDER
KADYAN
AIML,UPES

/R

MS

AA

AB



EXPERIMENT 1 :BRUTE FORCE TECHNIQUES

1. Sort a given set of elements using bubble and selection sort and hence find the time required to sort elements

Solution-

```
#include <stdio.h>

//function to swap
void swap(int *x, int *y);
//function to create an array
void create();
//function to print an array
void print(int arr[], int n);
//function to perform selection sort
void selection_sort(int arr[], int n);
//function to perform bubble sort
void bubble_sort(int arr[],int n);
//initialising main function
int main()
{
    //creating array
    create();
    return 0;
}

//taking input in a pointer
void swap(int *x, int *y)
{
    //performing swap with 3 variable
    int temp = *x;
    *x = *y;
    *y = temp;
}

void create()
{
    //variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    //initialisig array with length
    int arr[n];
    for (i = 0; i < n; i++)
    {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
    }
    printf("Enter what to perform\n");
    printf("1.)Selection sort\n");
    printf("2.)Bubble sort\n");
    scanf("%d",&i);
    //creating switch for selection and bubble
    //taking variable for calculating time
    clock_t start, end;
    //this will store the total time taken
    double cpu_time_used;
```

//creating switch for selection and bubble

```
switch(i)
{
case 1:
```

//initilising start function

```
start = clock();
```

//calling selection sort function

```
selection_sort(arr, n);
```

//calling print function after sort

```
print(arr, n);
```

//ending the process time taken

```
end = clock();
```

//computing total time taken

```
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
```

```
printf("Time taken: %lf seconds to perform Selection sort\n", cpu_time_used);
```

```
printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
```

```
break;
```

```
case 2:
```

//initilising start function

```
start = clock();
```

//calling selection sort function

```
bubble_sort(arr, n);
```

//calling print function after sort

```
print(arr, n);
```

//ending the process time taken

```
end = clock();
```

//computing total time taken

```
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
```

```
printf("Time taken: %lf seconds to perform Bubble sort\n", cpu_time_used);
```

```
printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
```

```
break;
```

```
}
```

//printing an array with input and length

```
void print(int arr[],int n)
```

```
{
```

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

```
{
```

```
printf("%d ", arr[i]);
```

```
}
```

```
printf("\n");
```

```
}
```

//taking input after creation of an array

```
void selection_sort(int arr[], int n)
```

```
{
```

//creating variables for minimum index and a counter for steps taken

```
int min_ind ,flag=0;
```

```
for (int i = 0; i < n - 1; i++)
```

```
{
```

```

//assuming first element is smallest(storing its address)
min_ind = i;
flag++;
for (int j = i + 1; j < n; j++)
{
    //comparing other elements of array form smallest
    if (arr[j] < arr[min_ind])
    {

//if element is smaller then storing new address for minimum
        min_ind = j;
    }
}

```

```

//checking if the prviously taken smallest
if (min_ind != i)
{
    //swaping the element with prvious smallest
    swap(&arr[min_ind], &arr[i]);
}
//printing counter
printf("\npass-%d\n", flag);
//printing array after each pass
print(arr, n);
}
}

```

```

void bubble_sort(int arr[], int n)
{
    //creating a variable counter
    int flag=0;
    for(int i=0; i<n-1; i++)
    {
        flag++;
        printf("\n phase: %d\n", flag);
        for(int j=0; j<n-1-i; j++)
        {
            //comparing first element with next
            if(arr[j]>arr[j+1])
            {
                //swap if first is smaller
                swap(&arr[j], &arr[j+1]);
            }
            else
            {
                //if not print the array as it is
                print(arr, n);
                continue;
            }
        }
        //print the array
        print(arr, n);
    }
}

```

BEST CASE OUTPUT-(SELECTION SORT)

```
PS C:\Users\ssris\Desktop\output\c\output> cd 'c:\Users\ssris\Desktop\output\c\output'
PS C:\Users\ssris\Desktop\output\c\output> & .\'Selection_sort.exe'
Enter the length of an array: 10
Enter the 1 element of an array: -10
Enter the 2 element of an array: -5
Enter the 3 element of an array: -2
Enter the 4 element of an array: 0
Enter the 5 element of an array: 1
Enter the 6 element of an array: 2
Enter the 7 element of an array: 3
Enter the 8 element of an array: 4
Enter the 9 element of an array: 7
Enter the 10 element of an array: 9
Enter what to perform
1.)Selection sort
2.)Bubble sort
1

pass-1
-10 -5 -2 0 1 2 3 4 7 9

pass-2
-10 -5 -2 0 1 2 3 4 7 9

pass-3
-10 -5 -2 0 1 2 3 4 7 9

pass-4
-10 -5 -2 0 1 2 3 4 7 9

pass-5
-10 -5 -2 0 1 2 3 4 7 9

pass-6
-10 -5 -2 0 1 2 3 4 7 9

pass-7
-10 -5 -2 0 1 2 3 4 7 9

pass-8
-10 -5 -2 0 1 2 3 4 7 9

pass-9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
Time taken: 0.006000 seconds to perform Selection sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
```

AVERAGE CASE OUTPUT-(SELECTION SORT)

```
PS C:\Users\ssris> cd 'c:\Users\ssris\Desktop\output\c\output'
PS C:\Users\ssris\Desktop\output\c\output> & .\'Selection_sort.exe'
Enter the length of an array: 10
Enter the 1 element of an array: 5
Enter the 2 element of an array: -3
Enter the 3 element of an array: 7
Enter the 4 element of an array: 1
Enter the 5 element of an array: 9
Enter the 6 element of an array: 2
Enter the 7 element of an array: 0
Enter the 8 element of an array: -6
Enter the 9 element of an array: 4
Enter the 10 element of an array: -10
Enter what to perform
1.)Selection sort
2.)Bubble sort
1

pass-1
-10 -3 7 1 9 2 0 -6 4 5

pass-2
-10 -6 7 1 9 2 0 -3 4 5

pass-3
-10 -6 -3 1 9 2 0 7 4 5

pass-4
-10 -6 -3 0 9 2 1 7 4 5

pass-5
-10 -6 -3 0 1 2 9 7 4 5

pass-6
-10 -6 -3 0 1 2 9 7 4 5

pass-7
-10 -6 -3 0 1 2 4 7 9 5

pass-8
-10 -6 -3 0 1 2 4 5 9 7

pass-9
-10 -6 -3 0 1 2 4 5 7 9
-10 -6 -3 0 1 2 4 5 7 9
Time taken: 0.008000 seconds to perform Selection sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output>
```

WORST CASE OUTPUT-(SELECTION SORT)

```
PS C:\Users\ssris\Desktop\output\c\output> & .\'Selection_sort.exe'
Enter the length of an array: 10
Enter the 1 element of an array: 9
Enter the 2 element of an array: 7
Enter the 3 element of an array: 4
Enter the 4 element of an array: -8
Enter the 5 element of an array: -9
Enter the 6 element of an array: -4
Enter the 7 element of an array: 1
Enter the 8 element of an array: 0
Enter the 9 element of an array: -5
Enter the 10 element of an array: -10
Enter what to perform
1.)Selection sort
2.)Bubble sort
1

pass-1
-10 7 4 -8 -9 -4 1 0 -5 9

pass-2
-10 -9 4 -8 7 -4 1 0 -5 9

pass-3
-10 -9 -8 4 7 -4 1 0 -5 9

pass-4
-10 -9 -8 -5 7 -4 1 0 4 9

pass-5
-10 -9 -8 -5 -4 7 1 0 4 9

pass-6
-10 -9 -8 -5 -4 0 1 7 4 9

pass-7
-10 -9 -8 -5 -4 0 1 7 4 9

pass-8
-10 -9 -8 -5 -4 0 1 4 7 9

pass-9
-10 -9 -8 -5 -4 0 1 4 7 9
-10 -9 -8 -5 -4 0 1 4 7 9
Time taken: 0.014000 seconds to perform Selection sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output> |
```

BEST CASE OUTPUT-(BUBBLE SORT)

```
PS C:\Users\ssris\Desktop\output\c\output> cd 'c:\Users\ssris\Desktop\
PS C:\Users\ssris\Desktop\output\c\output> & .\'Selection_sort.exe'
Enter the length of an array: 10
Enter the 1 element of an array: -10
Enter the 2 element of an array: -5
Enter the 3 element of an array: -2
Enter the 4 element of an array: 0
Enter the 5 element of an array: 1
Enter the 6 element of an array: 2
Enter the 7 element of an array: 3
Enter the 8 element of an array: 4
Enter the 9 element of an array: 7
Enter the 10 element of an array: 9
Enter what to perform
1.)Selection sort
2.)Bubble sort
2

phase: 1
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 2
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 3
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 4
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 5
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 6
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 7
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 8
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9

phase: 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
-10 -5 -2 0 1 2 3 4 7 9
Time taken: 0.059000 seconds to perform Bubble sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output>
```


AVERAGE CASE OUTPUT-(BUBBLE SORT)

```
Enter the length of an array: 10
Enter the 1 element of an array: 10
Enter the 2 element of an array: 9
Enter the 3 element of an array: 8
Enter the 4 element of an array: 7
Enter the 5 element of an array: 6
Enter the 6 element of an array: 5
Enter the 7 element of an array: 4
Enter the 8 element of an array: 3
Enter the 9 element of an array: 2
Enter the 10 element of an array: 1
Enter what to perform
1.)Selection sort
2.)Bubble sort
2

phase: 1
9 8 7 6 5 4 3 2 1 10

phase: 2
8 7 6 5 4 3 2 1 9 10

phase: 3
7 6 5 4 3 2 1 8 9 10

phase: 4
6 5 4 3 2 1 7 8 9 10

phase: 5
5 4 3 2 1 6 7 8 9 10

phase: 6
4 3 2 1 5 6 7 8 9 10

phase: 7
3 2 1 4 5 6 7 8 9 10

phase: 8
2 1 3 4 5 6 7 8 9 10

phase: 9
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Time taken: 0.007000 seconds to perform Bubble sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> |
```

WORST CASE OUTPUT-(BUBBLE SORT)

```
PS C:\Users\ssris\Desktop\output\c\output> & .\'Selection_sort.exe'
Enter the length of an array: 10
Enter the 1 element of an array: 5
Enter the 2 element of an array: 4
Enter the 3 element of an array: 3
Enter the 4 element of an array: 2
Enter the 5 element of an array: 1
Enter the 6 element of an array: 10
Enter the 7 element of an array: 9
Enter the 8 element of an array: 8
Enter the 9 element of an array: 7
Enter the 10 element of an array: 6
Enter what to perform
1.)Selection sort
2.)Bubble sort
2

phase: 1
4 3 2 1 5 10 9 8 7 6
4 3 2 1 5 9 8 7 6 10

phase: 2
3 2 1 4 5 9 8 7 6 10
3 2 1 4 5 9 8 7 6 10
3 2 1 4 5 8 7 6 9 10

phase: 3
2 1 3 4 5 8 7 6 9 10
2 1 3 4 5 8 7 6 9 10
2 1 3 4 5 8 7 6 9 10
2 1 3 4 5 7 6 8 9 10

phase: 4
1 2 3 4 5 7 6 8 9 10
1 2 3 4 5 7 6 8 9 10
1 2 3 4 5 7 6 8 9 10
1 2 3 4 5 7 6 8 9 10
1 2 3 4 5 6 7 8 9 10

phase: 5
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

phase: 6
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

phase: 7
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

phase: 8
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

phase: 9
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Time taken: 0.088000 seconds to perform Bubble sort
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output> S
```

2. Perform linear search and find the time required to search an element.

Solution-

```
#include <stdio.h>
#include <time.h>

int linearsearch(int arr[], int n);

int main() {
    //calling create function
    create();
}

void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i < n; i++) {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
    }
    // Initializing variables for calculating time
    clock_t start, end;
    // Initializing variables for calculating CPU time
    double cpu_time_used;
    // Starting the time measurement
    start = clock();
    // Calling the linearsearch function
    linearsearch(arr, n);
    end = clock();
    //ending the clock
    cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
    //printing the total time taken by cpu
    printf("Time taken is %0.3f seconds to execute.\n", cpu_time_used);
    printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
}

int linearsearch(int arr[], int n) {
    // Initializing counters, loop, and item variables
    int flag = 0, k = 0, i, item;
    //taking input of the element
    printf("\nEnter the element to be searched: ");
    scanf("%d", &item);
    for (i = 0; i < n; i++) {
        flag++;
        //comparing each element with the item need to be found
        if (arr[i] == item)
        {
            //adding variable to check whether element was present in the index or not

```

```

        //adding variable to check whether element was present in the index or not
        k++;
        //check for element in an array
        if(k==1)
        {
            printf("Element found at %d index.\n", i);
        }
        //check duplicate element in an array
        if(k!=0 && k!=1)
        {
            printf("Element also found at %d index.\n",i);
        }
        continue;
    }
}
//if element was not in the array then return with this exception
if (k == 0)
{
    printf("Element not found.\n");
}
//printing flag used to searched
printf("Flag: %d\n", flag);
}

```

SEARCHING WITHOUT ANY DUPLICATE ELEMENT

```

Enter the length of an array: 10
Enter the 1 element of an array: 10
Enter the 2 element of an array: 20
Enter the 3 element of an array: 30
Enter the 4 element of an array: 40
Enter the 5 element of an array: 50
Enter the 6 element of an array: 60
Enter the 7 element of an array: 70
Enter the 8 element of an array: 80
Enter the 9 element of an array: 90
Enter the 10 element of an array: 100

Enter the element to be searched: 50
Element found at 4 index.
Flag: 10
Time taken is 2.270 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```

SEARCHING WITH ANY DUPLICATE ELEMENT

```
Enter the length of an array: 10
Enter the 1 element of an array: 10
Enter the 2 element of an array: 10
Enter the 3 element of an array: 10
Enter the 4 element of an array: 10
Enter the 5 element of an array: 10
Enter the 6 element of an array: 10
Enter the 7 element of an array: 20
Enter the 8 element of an array: 20
Enter the 9 element of an array: 30
Enter the 10 element of an array: 30

Enter the element to be searched: 20
Element found at 6 index.
Element also found at 7 index.
Flag: 10
Time taken is 1.416 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>
```

3. Given a string called TEXT with 'n' characters and another string called PATTERN with 'm' characters ($m \leq n$). Write a program which implements brute force string matching to search for a given pattern in the text. If the pattern is present then find the position of first occurrences of Pattern in that Text.

SOLUTION-

```
#include<stdio.h>
#include<string.h>
void search(char* pat, char* txt)
{
    //Length of pattern string
    long p = strlen(pat);
    //length of text string
    long t = strlen(txt);
    //initializing loop till length of text -pattern
    for (int i = 0; i <= t - p; i++) {
        int j;

        //initializing loop till length of th pattern string
        for (j = 0; j < p; j++)
        {
            //checking condition for pattern in text
            if ( txt[i + j] != pat[j] )
                break;
        }
        if ( j == p ) //if the j is initialized in full then check for p
        {
            printf("Pattern found at index %d \n", i);
        }
    }
}

int main(void)
{
    char txt[] = "Hi mera naam rishabh sharma hai";
    char pat[] = "rishab";
    //calling function for searching pattern
    search(pat, txt);
}
```

OUTPUT-

```
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-bi0npyvr.anr' '--stdout=Microsoft-MIEngine-Out-1ra1pbze.0e1' '--stderr=Microsoft-MIEngine-Error-15ucs2e.141' '--pid=Microsoft-MIEngine-Pid-3hvwfkxb.d00' '--dbgExe=C:\i686-8.1.0-release-win32-dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--interpreter=mi'
Pattern found at index 13
PS C:\Users\ssris> 
```

EXPERIMENT 2 : DIVIDE AND CONQUER-I

1. Implement Binary search and linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.

Solution-

```
#include <stdio.h>
#include <time.h>
//function to create an array
void create();
//function to perform linear search
void linear_search(int arr[], int n);
//function to perform binary search
void binary_search(int arr[],int n);
//initialising main function
int main()
{
    //creating array
    create();
    return 0;
}

void create()
{
    //variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    //initialisig array with length
    int arr[n];
    for (i = 0; i < n; i++)
    {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
    }
    printf("Enter what to perform\n");
    printf("1.)Linear search\n");
    printf("2.)Binary search\n");
    scanf("%d",&i);
    //taking variable for calculating time
    clock_t start, end;
    //this will store the total time taken
    double cpu_time_used;
    //creating switch for selection and bubble
    switch(i)
    {
        case 1:
            //initilising start function
            start = clock();
            //calling linear search function
            linear_search(arr, n);
            //ending the process time taken
            end = clock();
            //computing total time taken
            cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
            printf("Time taken: %lf seconds to perform linear search\n", cpu_time_used);
            printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
            break;

        case 2:
            //initilising start function
```

```

start = clock();
//calling selection sort function
binary_search(arr, n);
//ending the process time taken
end = clock();
//computing total time taken
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
printf("Time taken: %lf seconds to perform Binary search\n", cpu_time_used);
printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
break;

}

}

//taking input after creation of an array
void linear_search(int arr[], int n)
{
    // Initializing counters, loop, and item variables
    int flag = 0, k = 0, i, item;
    //taking input of the element
    printf("\nEnter the element to be searched: ");
    scanf("%d", &item);
    for (i = 0; i < n; i++) {
        flag++;
        //comparing each element with the item need to be found
        if (arr[i] == item)
        {
            //adding variable to check whether element was present in the index or not
            k++;
            //check for element in an array
            if(k==1)
            {
                printf("Element found at %d index.\n", i);
            }
            //check duplicate element in an array
            if(k!=0 && k!=1)
            {
                printf("Element also found at %d index.\n",i);
            }

            continue;
        }
    }
    //if element was not in the array then return with this exception
    if (k == 0)
    {
        printf("Element not found.\n");
    }
    //printing flag used to searched
    printf("Flag: %d\n", flag);
}

void binary_search(int arr[],int n)
{
    // Initializing counters, loop, and item variables
    int flag=0,item,mid,beg,end;
    //initializing end variable
    end=n-1;
    //initializing start variable

```



```

beg=0;
printf("Enter the number to be searched in an array : \n");
//getting value of item
scanf("%d",&item);
//initilizing loop with condition
while(beg<=end)
{
    flag++;
//initilizing mid variable
    mid=(beg+end)/2;
//comparing item with mid
    if(item==arr[mid])
    {
        printf("Element Found at %d\n",mid);
        break;
    }
//condition if item is greater than mid of an array
    else if(item>arr[mid])
    {
        beg=mid+1;
    }
//condition if item is smaller than mid of an array
    else
    {
        end=mid-1;
    }
}
}
}

```

LINEAR SEARCH

```

Enter the length of an array: 5
Enter the 1 element of an array: 10
Enter the 2 element of an array: 25
Enter the 3 element of an array: 30
Enter the 4 element of an array: 45
Enter the 5 element of an array: 50
Enter what to perform
1.)Linear search
2.)Binary search
1

Enter the element to be searched: 25
Element found at 1 index.
Flag: 5
Time taken: 12.015000 seconds to perform linear search
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> █

```

BINARY SEARCH

```
Enter the length of an array: 5
Enter the 1 element of an array: 10
Enter the 2 element of an array: 30
Enter the 3 element of an array: 50
Enter the 4 element of an array: 70
Enter the 5 element of an array: 90
Enter what to perform
1.)Linear search
2.)Binary search
2
Enter the number to be searched in an array :
90
Element Found at 4
Flag: 3
Time taken: 2.441000 seconds to perform Binary search
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>
```

2. Search elements using the Binary search method and determine the time required to search the element. Repeat the experiment for different values of n , to search for the element in the list and plot a graph of the time taken versus n .

Solution:

```
#include <stdio.h>
#include <time.h>

int main(void)
{
    // Initializing clock
    clock_t start, endt;
    double cpu_time_used;
    start = clock();

    int i, n;
    // Taking input length of the array
    printf("Enter the length of an array: ");
    scanf("%d", &n);

    // Initializing the array
    int arr[n];

    // Taking input for the array
    for (i = 0; i < n; i++)
    {
        printf("Enter the value at %d: ", i);
        scanf("%d", &arr[i]);
    }

    // Initializing counters, loop, and item variables
    int flag = 0, item, mid, beg, end;

    // Initializing end variable
    end = n - 1;
    // Initializing start variable
    beg = 0;

    printf("Enter the number to be searched in the array: ");
    // Getting the value of the item
    scanf("%d", &item);

    i = 0; // counter

    // Initializing loop with condition
    while (beg <= end)
    {
        flag++;

        // Initializing mid variable
        mid = (beg + end) / 2;

        // Comparing item with mid
        if (item == arr[mid])
        {
            i++;
            printf("Element Found at %d\n", mid);
            break;
        }

        // Condition if item is greater than mid of an array
        else if (item > arr[mid])
        {
            beg = mid + 1;
```

```

}
// Condition if item is smaller than mid of an array
else
{
    end = mid - 1;
}
}

// Checking if the element is present or not
if (i != 1)
{
    printf("Element not found in the array\n");
}

// Printing the flag used for the search
printf("Flag: %d\n");
}

```

BINARY SEARCH-

```

enter the length of an array7
Enter the value at 0 :25
Enter the value at 1 :50
Enter the value at 2 :75
Enter the value at 3 :100
Enter the value at 4 :125
Enter the value at 5 :150
Enter the value at 6 :175
Enter the number to be searched in an array :
25
Element Found at 0
Flag: 3
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\debugAdapters\bin\WindowsDebugL
n=Microsoft-MIEngine-In-kh15a1np.13j' '--stdout=Micro
fp0ejfc.nqn' '--stderr=Microsoft-MIEngine-Error-ueqr
rosoft-MIEngine-Pid-flptmljo.ge2' '--dbgExe=C:\i686-
dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--interpreter

enter the length of an array6
Enter the value at 0 :10
Enter the value at 1 :20
Enter the value at 2 :30
Enter the value at 3 :40
Enter the value at 4 :50
Enter the value at 5 :60
Enter the number to be searched in an array :
1
Element not found in the array
Flag: 2
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```

3.Sort a given set of elements using the Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Solution-

```
#include <stdio.h>
#include<time.h>
void create();
void print(int arr[],int n);

void merge(int arr[], int p, int q, int r) {
    //getting left half element in n1
    int n1 = q - p + 1;
    //getting right half elements in n2
    int n2 = r - q;
    //creating two arrays
    int L[n1], M[n2];

    for (int i = 0; i < n1; i++)
        //storing elements in left array those are below mid
        L[i] = arr[p + i];
    for (int j = 0; j < n2; j++)
        //storing elemnts in right array those are above mid
        M[j] = arr[q + 1 + j];

    int i, j, k;
    //counter variable for the left array
    i = 0;
    //counter variable for the right array
    j = 0;
    //counter variable for sorted array
    k = p;
    //checking if any of the one array is empty
    while (i < n1 && j < n2) {
        if (L[i] <= M[j])
            //comparing if the left element is small of right
            {
                arr[k] = L[i];
                i++;
            }
        else
            //if not then add right element to sorted array
            {
                arr[k] = M[j];
                j++;
            }
        k++;
    }
    //if elements are left in left array
    while (i < n1) {
        //directly add it to k(SORTED ARRAY)
        arr[k] = L[i];
        i++;
        k++;
    }

    //if elements are left in right array
    while (j < n2) {
        //aad it to k (SORTED ARRAY)
        arr[k] = M[j];
```

```

j++;
k++;
}
}

void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        //initializing a variable for the mid term
        int m = l + (r - l) / 2;
        //left side partition of the array
        mergeSort(arr, l, m);
        //right side partition of the array
        mergeSort(arr, m + 1, r);
        //after dividing the array the elements are now sorted and merged
        merge(arr, l, m, r);
    }
}

void print(int arr[], int n)
{
    //printing array with n elements
    for (int i=0 ; i <= n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    create();
    return 0;
}

void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i < n; i++) {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
    }
    // Initializing variables for calculating time
    clock_t start, end;
    // Initializing variables for calculating CPU time
    double cpu_time_used;
    // Starting the time measurement
    start = clock();
    mergeSort(arr, 0, n - 1);
    //ending the clock
    printf("The sorted array after applying merge is-\n");
    //printing the sorted array
    print(arr, n);
    end = clock();
    //printing the total time taken by cpu
    cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("Time taken is %0.3f seconds to execute.\n", cpu_time_used);
    printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
}

```

WORST CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: 8
Enter the 2 element of an array: 7
Enter the 3 element of an array: 6
Enter the 4 element of an array: 5
Enter the 5 element of an array: 4
Enter the 6 element of an array: 3
Enter the 7 element of an array: 2
Enter the 8 element of an array: 1
The sorted array after applying merge is-
1 2 3 4 5 6 7 8
Time taken is 0.003 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>
```

BEST CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: 1
Enter the 2 element of an array: 2
Enter the 3 element of an array: 3
Enter the 4 element of an array: 4
Enter the 5 element of an array: 5
Enter the 6 element of an array: 6
Enter the 7 element of an array: 7
Enter the 8 element of an array: 8
The sorted array after applying merge is-
1 2 3 4 5 6 7 8
Time taken is 0.001 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output>
```

AVERAGE CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: 7
Enter the 2 element of an array: 2
Enter the 3 element of an array: 8
Enter the 4 element of an array: 4
Enter the 5 element of an array: 9
Enter the 6 element of an array: 1
Enter the 7 element of an array: 5
Enter the 8 element of an array: 0
The sorted array after applying merge is-
0 1 2 4 5 7 8 9
Time taken is 0.002 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris\Desktop\output\c\output>
```

With the help of rand() function we can generate random input and can also sort it with the same algorithm logic

First we need to do some change in code in create() function

```
void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i < n; i++) {
        //for random input in the code
        arr[i]=rand()%100;
    }
    printf("unsorted array input with rand() function\n");
    print(arr,n);
}
```

OUTPUT WITH THE HELP OF RAND() FUNCTION-

```
Enter the length of an array: 10
unsorted array input with rand() function
41 67 34 0 69 24 78 58 62 64
The sorted array after applying merge is-
0 24 34 41 58 62 64 67 69 78
Time taken is 0.001 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extension
ters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-M
Engine-Out-imelqnf4.gln' '--stderr=Microsoft-MIEngine-E
d-p2ccaki3.jqm' '--dbgExe=C:\i686-8.1.0-release-win32-d
eter=mi'
Enter the length of an array: 7
unsorted array input with rand() function
41 67 34 0 69 24 78
The sorted array after applying merge is-
0 24 34 41 67 69 78
Time taken is 0.002 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extension
ters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-M
Engine-Out-xmgoggal.psd' '--stderr=Microsoft-MIEngine-E
d-oae4s1dx.qya' '--dbgExe=C:\i686-8.1.0-release-win32-d
eter=mi'
Enter the length of an array: 4
unsorted array input with rand() function
41 67 34 0
The sorted array after applying merge is-
0 34 41 67
Time taken is 0.001 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>
```


EXPERIMENT 3 : DIVIDE AND CONQUER-II

1.Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

SOLUTION-

```
#include <stdio.h>
#include<time.h>
void swap(int *x, int *y);
void create();
void print(int arr[],int x,int n);
void quick_sort(int arr[], int first,int last);

int main() {
    create();
    return 0;
}

//taking input in a pointer
void swap(int *x, int *y)
{
    //performing swap with 3 variable
    int temp = *x;
    *x = *y;
    *y = temp;
}

void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i < n; i++) {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
    }
    // Initializing variables for calculating time
    clock_t start, end;
    // Initializing variables for calculating CPU time
    double cpu_time_used;
    // Starting the time measurement
    start = clock();
    int first=0;
    int last =n-1;
    quick_sort(arr,first,last);
    //ending the clock
    printf("The sorted array after applying quick sort is-\n");
    //printing the sorted array
    print(arr,first,last);
    end = clock();
    //printing the total time taken by cpu
    cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("Time taken is %0.3f seconds to execute.\n", cpu_time_used);
    printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
}
```

```

void print(int arr[],int x,int n)
{
    //printing array with n elements
    for (int i=x ; i <= n-1; i++)

    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void quick_sort(int arr[ ],int first ,int last)
{ //creating intilising variable for sorting
    int i,j,pivot;
    //checking the condition of empty array
    if(first<last){
        //initializing the variables like pivot left and right
        pivot=first;
        i=first;
        j=last;
        //will initialize loop for the swapping
        while(i<j){
            //checking conditon to run left variable
            while(arr[i]<=arr[pivot] && i<last)
                i++;
            //checking conditon to run right variable
            while(arr[j]>arr[pivot])
                j--;
            //checking if i and j stops and i<j then swap
            if(i<j)
            {
                //swap the i element with j
                swap(&arr[i],&arr[j]);
            }
        }

        //swap the pivot element with j
        swap(&arr[pivot],&arr[j]);
        //applying the quick sort on left side of divided array
        quick_sort(arr,first,j-1);
        //applying the quick sort on right side of divided array
        quick_sort(arr,j+1,last);
    }
}

```

OUTPUT WHEN ALL ELEMENTS ARE SIMILAR IN QUICK SORT-

```

Enter the length of an array: 8
Enter the 1 element of an array: 1
Enter the 2 element of an array: 1
Enter the 3 element of an array: 1
Enter the 4 element of an array: 1
Enter the 5 element of an array: 1
Enter the 6 element of an array: 1
Enter the 7 element of an array: 1
Enter the 8 element of an array: 1
1 1 1 1 1 1 1 1
The sorted array after applying quick sort is-
1 1 1 1 1 1 1 1
Time taken is 0.001000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```

SORTED DESENDING CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: 8
Enter the 2 element of an array: 7
Enter the 3 element of an array: 6
Enter the 4 element of an array: 5
Enter the 5 element of an array: 4
Enter the 6 element of an array: 3
Enter the 7 element of an array: 2
Enter the 8 element of an array: 1
8 7 6 5 4 3 2 1
The sorted array after applying quick sort is-
1 2 3 4 5 6 7 8
Time taken is 0.001000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>
```

SORTED ASSENDING CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: 1
Enter the 2 element of an array: 2
Enter the 3 element of an array: 3
Enter the 4 element of an array: 4
Enter the 5 element of an array: 5
Enter the 6 element of an array: 6
Enter the 7 element of an array: 7
Enter the 8 element of an array: 8
1 2 3 4 5 6 7 8
The sorted array after applying quick sort is-
1 2 3 4 5 6 7 8
Time taken is 0.002000 seconds to execute.
```

RANDOM INPUT CASE OUTPUT-

```
Enter the length of an array: 8
Enter the 1 element of an array: -9
Enter the 2 element of an array: -7
Enter the 3 element of an array: 4
Enter the 4 element of an array: 0
Enter the 5 element of an array: 8
Enter the 6 element of an array: 88
Enter the 7 element of an array: 1
Enter the 8 element of an array: 6
-9 -7 4 0 8 88 1 6
The sorted array after applying quick sort is-
-9 -7 0 1 4 6 8 88
Time taken is 0.001000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
```

With the help of rand() function we can generate random input and can also sort it with the same algorithm logic

First we need to do some change in code in create() function

```
void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i <= n; i++) {
        /*
            printf("Enter the %d element of an array: ", i + 1);
            scanf("%d", &arr[i]);
        */
        //applying random function for random inputs
        arr[i]=rand()%100+99;
    }
}
```

```

Enter the length of an array: 8
-58 -32 -65 -99 -30 -75 -21 -41 -37
The sorted array after applying quick sort is-
-99 -75 -65 -58 -41 -37 -32 -30 -21
Time taken is 0.001000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-vscode.cpptools\bin\WindowsDebugLauncher.exe' '-.
.2u0' '--stdout=Microsoft-MIEngine-Out-dccjddxo.aoc
chxisbb0.tgp' '--pid=Microsoft-MIEngine-Pid-gcblx4d
e-win32-dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--in
Enter the length of an array: 5
140 166 133 99 168 123
The sorted array after applying quick sort is-
99 123 133 140 166 168
Time taken is 0.001000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-vscode.cpptools\bin\WindowsDebugLauncher.exe' '-.
.avc' '--stdout=Microsoft-MIEngine-Out-d4quxyut.crh
zoi52b5q.q05' '--pid=Microsoft-MIEngine-Pid-245tsml
e-win32-dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--in
Enter the length of an array: 10
140 166 133 99 168 123 177 157 161 163 104
The sorted array after applying quick sort is-
99 104 123 133 140 157 161 163 166 168 177
Time taken is 0.004000 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```

2.Sort a given set of elements using the insertion Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

SOLUTION-

```

#include <stdio.h>
#include<time.h>
void create();
void print(int arr[],int n);

void print(int arr[],int n)
{
    //printing array with n elements
    for (int i=0 ; i <n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void insertionSort(int array[], int n) {
    //initilizing loop variable
    for (int i= 1; i < n; i++) {
        //initializing temp variable with i+1 value
        int temp = array[i];
        //initializing j with 0
        int j = i - 1;
        //condition check if temp value is greater smaller than last j value
        while (j>=0 && array[j]>temp) {
            array[j + 1] = array[j];
            //if the temp is more smaller it will be compared with next least value of j
            j--;
        }
        array[j + 1] = temp;
    }
}

```

```

    }
    //if temp value is greater than value at j it will directly be added to sorted subarray
    array[j + 1] = temp;
    }
}
int main() {
    create();
}
void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i <= n-1; i++) {
        printf("Enter the %d element of an array: ", i + 1);
        scanf("%d", &arr[i]);
        //arr[i]=rand()%100;
    }
    // Initializing variables for calculating time
    clock_t start, end;
    // Initializing variables for calculating CPU time
    double cpu_time_used;
    // Starting the time measurement
    start = clock();
    print(arr,n);
    insertionSort(arr,n);
    //ending the clock
    printf("The sorted array after applying insertion sort is-\n");
    //printing the sorted array
    print(arr,n);
    end = clock();
    //printing the total time taken by cpu
    cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("Time taken is %0.3f seconds to execute.\n", cpu_time_used);
    printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
}

```

OUTPUT WITH RANDOM INPUT-

```

Enter the length of an array: 6
Enter the 1 element of an array: 5
Enter the 2 element of an array: 4
Enter the 3 element of an array: 10
Enter the 4 element of an array: 1
Enter the 5 element of an array: 6
Enter the 6 element of an array: 2
5 4 10 1 6 2
The sorted array after applying insertion sort is-
1 2 4 5 6 10
Time taken is 0.003 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```

OUTPUT WITH WORST CASE-

```

Enter the length of an array: 8
Enter the 1 element of an array: 8
Enter the 2 element of an array: 7
Enter the 3 element of an array: 6
Enter the 4 element of an array: 5
Enter the 5 element of an array: 4
Enter the 6 element of an array: 3
Enter the 7 element of an array: 2
Enter the 8 element of an array: 1
8 7 6 5 4 3 2 1
The sorted array after applying insertion sort is-
1 2 3 4 5 6 7 8
Time taken is 0.004 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA

```

OUTPUT WITH BEST CASE-

```

Enter the length of an array: 8
Enter the 1 element of an array: 1
Enter the 2 element of an array: 2
Enter the 3 element of an array: 3
Enter the 4 element of an array: 4
Enter the 5 element of an array: 5
Enter the 6 element of an array: 6
Enter the 7 element of an array: 7
Enter the 8 element of an array: 8
1 2 3 4 5 6 7 8
The sorted array after applying insertion sort is-
1 2 3 4 5 6 7 8
Time taken is 0.003 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA

```

OUTPUT WITH AVERAGE CASE-

```

Enter the length of an array: 8
Enter the 1 element of an array: -9
Enter the 2 element of an array: 9
Enter the 3 element of an array: -8
Enter the 4 element of an array: 8
Enter the 5 element of an array: -2
Enter the 6 element of an array: 2
Enter the 7 element of an array: 0
Enter the 8 element of an array: 7
-9 9 -8 8 -2 2 0 7
The sorted array after applying insertion sort is-
-9 -8 -2 0 2 7 8 9
Time taken is 0.004 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA

```

With the help of rand() function we can generate random input and can also sort it with the same algorithm logic

First we need to do some change in code in create() function

```

void create() {
    // Variable for length and loop
    int i, n;
    printf("Enter the length of an array: ");
    scanf("%d", &n);
    // Initializing the array with the given length
    int arr[n];
    for (i = 0; i <= n-1; i++) {
        //printf("Enter the %d element of an array: ", i + 1);
        //scanf("%d", &arr[i]);
        arr[i]=rand()%100;
    }
}

```



```

Enter the length of an array: 8
41 67 34 0 69 24 78 58
The sorted array after applying insertion sort is-
0 24 34 41 58 67 69 78
Time taken is 0.003 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-
-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Mi
.ol4' '--stdout=Microsoft-MIEngine-Out-nf4ty4w2.r4x' '--stde
qtdnfub2.dg3' '--pid=Microsoft-MIEngine-Pid-1u1qtjpa.v2q' '-
e-win32-dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--interpreter
Enter the length of an array: 5
41 67 34 0 69
The sorted array after applying insertion sort is-
0 34 41 67 69
Time taken is 0.002 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris> & 'c:\Users\ssris\.vscode\extensions\ms-
-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Mi
.2bx' '--stdout=Microsoft-MIEngine-Out-cxwoux4a.owd' '--stde
tjo4bc1r.xww' '--pid=Microsoft-MIEngine-Pid-4e55jyyk.xe2' '-
e-win32-dwarf-rt_v6-rev0\mingw32\bin\gdb.exe' '--interpreter
Enter the length of an array: 20
41 67 34 0 69 24 78 58 62 64 5 45 81 27 61 91 95 42 27 36
The sorted array after applying insertion sort is-
0 5 24 27 27 34 36 41 42 45 58 61 62 64 67 69 78 81 91 95
Time taken is 0.006 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA

```

3.Implement Strassen's matrix multiplication and compare the complexity with normal matrix multiplication

SOLUTION-

```

#include<stdio.h>
#include<time.h>
int main(){
    // Initializing variables for calculating time
    clock_t start, end;
    // Initializing variables for calculating CPU time
    double cpu_time_used;
    // Starting the time measurement
    start = clock();
    //initilizing an array for final solution
    int z[2][2];
    //taking variables for loops in 2 X 2 matrix
    int i, j;
    //initilizing variables for equations
    int m1, m2, m3, m4 , m5, m6, m7;
    //defining the values of 1st array
    int x[2][2] = {
        {112, 834},
        {222, 150}
    };
    //defining the values of 2nd array
    int y[2][2] = {
        {3, 46},
        {52, 81}
    };
    //printing the values of 1st array
    printf("\nThe first matrix is\n");
    for(i = 0; i < 2; i++) {
        printf("\n");
        for(j = 0; j < 2; j++)
            printf("%d\t", x[i][j]);
    }
}

```

```

}

//printing the values of 2nd array
printf("\nThe second matrix is\n");
for(i = 0; i < 2; i++) {
    printf("\n");
    for(j = 0; j < 2; j++)
        printf("%d\t", y[i][j]);
}

//defining values of the equation of stressen
m1= (x[0][0] + x[1][1]) * (y[0][0] + y[1][1]);
m2= (x[1][0] + x[1][1]) * y[0][0];
m3= x[0][0] * (y[0][1] - y[1][1]);
m4= x[1][1] * (y[1][0] - y[0][0]);
m5= (x[0][0] + x[0][1]) * y[1][1];
m6= (x[1][0] - x[0][0]) * (y[0][0]+y[0][1]);
m7= (x[0][1] - x[1][1]) * (y[1][0]+y[1][1]);
z[0][0] = m1 + m4- m5 + m7;
z[0][1] = m3 + m5;
z[1][0] = m2 + m4;
z[1][1] = m1 - m2 + m3 + m6;

//printing the values after product from z array
printf("\nProduct achieved using Strassen's algorithm \n");
for(i = 0; i < 2 ; i++) {
    printf("\n");
    for(j = 0; j < 2; j++)
        printf("%d\t", z[i][j]);
}
printf("\n");
end = clock();

//printing the total time taken by cpu
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
printf("Time taken is %0.3f seconds to execute.\n", cpu_time_used);
printf("Jitna dimaag tha utna laga diya -BY RISHABH SHARMA");
return 0;
}

```

OUTPUT OF STRESSEN'S MATRIX MULTIPLICATION-

```

The first matrix is
112      834
222      150
The second matrix is
3         46
52        81
Product achieved using Strassen's algorithm

43704    72706
8466     22362
Time taken is 0.002 seconds to execute.
Jitna dimaag tha utna laga diya -BY RISHABH SHARMA
PS C:\Users\ssris>

```