

Insertion Sort

Topic
#include <stdio.h>
int main()

```

{
    int i, j, array[100], temp, n;
    printf("Enter the no. of elements in array: ");
    scanf("%d", &n);
    printf("Enter the values of the array: ");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &array[i]);
    }
    for (i = 1; i < n; i++)
    {
        temp = array[i];
        j = i - 1;
        while (j >= 0 & array[j] > temp)
        {
            array[j+1] = array[j];
            j = j - 1;
        }
        array[j+1] = temp;
    }
    printf("Elements after sorting:\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\n", array[i]);
    }
    return 0;
}

```

}

Ass - II

Q. Given an array. Let us assume that there can be duplicates in the list. Write a program to search for an element in the list in such a way that we get the highest index if there are duplicates.

Input

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

```
int main()
```

```
{
```

```
int i, j, n, arr[50], occur=0, store=0;
```

```
printf("Enter the number of elements: \n");
```

```
scanf("%d", &n);
```

```
printf("Enter the values: \n");
```

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

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

```
}
```

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

```
for (j=i+1; j<n; j++){
```

```
if (arr[i] == arr[j])
```

```
{
```

```
occur++;
```

```
store = j;
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
printf("Number of duplicates occurred: %d\n", occur);
```

```
printf("Highest Index value is: %d\n", store);
```

```
return 0;
```

```
}
```

Output

Elements:

② Write a program for finding i and j in an array A for any key such that $A[j]^2 + A[i]^2 = \text{key}$

sol Input

```
#include <stdio.h>
int main()
```

```
{
```

```
int n, arr[50], i, j, key;
```

```
printf("Enter the array size: ");
```

```
scanf("%d", &n);
```

```
printf("Enter the elements: \n");
```

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

```
{
```

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

```
}
```

```
printf("Enter the key: ");
```

```
scanf("%d", &key);
```

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

```
{
```

```
for (j=i+1; j<n; j++)
```

```
{
```

```
if (arr[i]*arr[i] + arr[j]*arr[j] == key)
```

```
{
```

```
printf("pair found (%d %d)\n", i, j);
```

```
return 0;
```

```
}
```

```
}
```

```
}
```

Output

Enter the array size: 5

Enter the elements:

3

4

8

4

2

Enter the key: 25

Given key in a sorted array A with distinct values. What is the index of key such that $A[i] + A[j] + A[k]$ is equal to key.

int main() {
int n;
int arr[n];
int key;
int i, j, k;
int sum;
int found = 0;
return 0;
}

```
int main() {
    int n, arr[n], key, count = 0;
    printf("Enter the no. of array: ");
    scanf("%d", &n);
    printf("Enter the elements: \n");
    for (i = 0; i < n; i++) {
        printf("%d", arr[i]);
        scanf("%d", &arr[i]);
    }
    printf("Enter the key: ");
    scanf("%d", &key);
    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            for (k = j + 1; k < n; k++) {
                if (arr[i] + arr[j] + arr[k] == key) {
                    printf("Pair found (%d, %d, %d) in", arr[i], arr[j], arr[k]);
                    count++;
                }
            }
        }
    }
    if (count == 0) {
        printf("No pair found");
    }
}
```

```
int main() {
    int n, arr[n], key, count = 0;
    printf("Enter the no. of array: ");
    scanf("%d", &n);
    printf("Enter the elements: \n");
    for (i = 0; i < n; i++) {
        printf("%d", arr[i]);
        scanf("%d", &arr[i]);
    }
    printf("Enter the key: ");
    scanf("%d", &key);
    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            for (k = j + 1; k < n; k++) {
                if (arr[i] + arr[j] + arr[k] == key) {
                    printf("Pair found (%d, %d, %d) in", arr[i], arr[j], arr[k]);
                    count++;
                }
            }
        }
    }
    if (count == 0) {
        printf("No pair found");
    }
}
```

We are going to implement GCD using the following 3 algorithms:
 1. Euclidean algorithm
 2. Binary GCD algorithm
 3. Common prime factors method

Input
 Enter the two numbers: 1414 2
 The gcd of the two numbers is 2

int a, b, num;
 printf ("Enter the two numbers: ");
 scanf ("%d %d", &a, &b);
 while (b != 0)
 {
 num = a % b;
 a = b;
 b = num;
 }
 printf ("The gcd of the two numbers is %d", a);
 return 0;

Input
 Enter the two numbers: 31415
 The gcd of the two numbers is 1

#include <stdio.h>
 int HCF (int a, int b)

int a, b, num;
 printf ("Enter the two numbers: ");
 scanf ("%d %d", &a, &b);
 num = HCF (a, b);
 printf ("The gcd of the two numbers is %d", num);

③

Input

include <stdio.h>
int HCF (int n1, n2)

int n3, 1, ans = 1, n2 = 0, m2 = 0;
for (i = 2; i <= n3; i++)

n2 = 0;
if (n1 % i == 0)

n1 = 1;
if (n2 % i == 0)

m2 = n2 / i;
n2 = m2 / i;

Output

Enter the two numbers : 36 45
The gcd of the two number is 1

else
{
for (i = 2; i <= n3; i++)
{
if (n1 % i == 0 && n2 % i == 0)

return (i);
}

return 1;

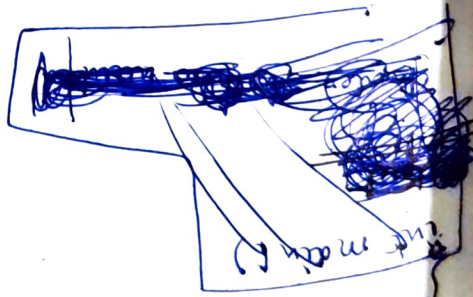
if (a < b)
int i = c;

int HCF (int a, int b)

Output
Enter the two numbers: 31415 14142
The gcd of two numbers is 1.

int m1, n2, gcd;
printf("Enter the two numbers: ");
scanf("%d %d", &m1, &n2);
gcd = HCF(m1, n2);
printf("The gcd of the two numbers is %d", gcd);

return ans;



if (n1 & n2) {
ans = ans * n1;
if (n1 & n2) {
return ans;
}
}
if (n1 & n2) {
return ans;
}
}

Write a program to sort an array using merge sort.

Answer:-

```
void mergeSort(int a[], int i, int j)
{
    if (i < j)
    {
        int mid = (i+j)/2;
        mergeSort(a, i, mid);
        mergeSort(a, mid+1, j);
        merge(a, i, mid, j);
    }
}
```

```
void merge(int a[], int i1, int i2, int j1, int j2)
{
    // ...
}
```

```
int temp[50];
int i1, i2, j1, j2;
k = 0;
u = i2;
v = j1;
while (i1 <= j1 & v <= j2)
{
    if (a[i1] < a[v])
    {
        temp[k++] = a[i1++];
    }
    else
    {
        temp[k++] = a[v++];
    }
}
```

```
while (i1 <= j1)
{
    temp[k++] = a[i1++];
}
while (v <= j2)
{
    temp[k++] = a[v++];
}
```


9
7
5
3
1

Enter no of elements : 5
Enter array elements : 8

Output

```

    }
    return 0;
}

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

printf("\n Sorted array is : ");
mergeSort(a, 0, n-1);

```

```

    scanf("%d", &a[i]);
}

for (i = 0; i < n; i++) {
    printf("Enter array elements: ");
    scanf("%d", &n);
    printf("Enter the number of elements: ");
    int a[30], n, i;
}

```

```

int main()
{

```

```

    a[i] = temp[i];
}

```

```

for (i = 1; i <= n/2; i++) {
    for (j = 2*i; j <= n; j++) {

```

```
int i;
for (i=0; i<n; i++)
    printf("%d", a[i]);
```

```
void PrintArr (int a[], int n)
```

```
quick (a, p+1, end);
```

```
int p = partition (a, start, end);
```

```
if (start < end)
```

```
void quick (int a[], int start, int end)
```

```
return (i+1);
```

```
a[end] = t;
```

```
a[i+1] = a[end];
```

```
int t = a[i+1];
```

```
++i;
int t = a[i];
a[i] = a[j];
a[j] = t;
```

```
if (a[j] < pivot)
```

```
for (int j = start; j <= end-1; j++)
```

```
int pivot = a[end];
int i = [start-1];
```

```
partition (int a[], int start, int end)
```

int main()

int a[50], m, i;

printf("Enter the no. of elements: ");

scanf("%d", &m);

printf("Enter array elements: ");

for (i = 0; i < m; i++)

scanf("%d", &a[i]);

printf("Before Sorting array elements are: ");

printf(a, m);

printf(a, m);

quick(a, 0, m-1);

printf("After Sorting array elements are: ");

printf(a, m);

printf(a, m);

return 0;

Enter the no. of elements: 5

Enter array elements: 10 4 12 8 6

Before sorting array elements are:

10 4 12 8 6

After sorting array elements are:

4 6 8 10 12

QuickSort with random function

Input

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

int partition (int *arr, int s, int f)

```

    srand (time (0));
    int p = s + rand () % (f - s + 1);
    int pivot = arr[p];
    arr[s] = arr[p];
    arr[p] = pivot;
    int i = s;
    for (int j = f; j > s; j--)
        if (arr[j] > pivot)
            swap (arr[j], arr[i]);
    swap (arr[i], arr[p]);
    return i;

```

void quicksort (int *arr, int s, int f)

```

    if (s >= f)
        return;
    int p = partition (arr, s, f);
    quicksort (arr, s, p-1);
    quicksort (arr, p+1, f);

```

int main()

```

    int n;
    printf ("Enter the size of the array: ");
    scanf ("%d", &n);
    int *arr = (int *) malloc (size of (int) * n);

```