**Question 01**

#include <stdio.h>

#include <string.h>

void sortNumericArray(int arrnum[], int n)

{

int i = 0, j = 0, temp;

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

{

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

{

if (arrnum[i] > arrnum[j])

{

temp = arrnum[i];

arrnum[i] = arrnum[j];

arrnum[j] = temp;

}

}

}

}

void sortAlphabeticalArray(char arrchar[], int m)

{

int i = 0, j = 0;

char temp;

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

{

for (j = i + 1; j < m; j++)

{

if (arrchar[i] > arrchar[j])

{

temp = arrchar[i];

arrchar[i] = arrchar[j];

arrchar[j] = temp;

}

}

}

}

int main()

{

int n, m, choice, i;

char arrchar[1000];

printf("Enter:\n1 for a numeric array\n2 for a character array\n");

scanf("%d", &choice);

if (choice == 1)

{

printf("How many elements do you want in your numeric array?\n");

scanf("%d", &n);

int arrnum[n];

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

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

{

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

}

printf("The elements of the numeric array sorted in ascending order are as follows\n");

sortNumericArray(arrnum, n);

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

{

printf("%d\n", arrnum[i]);

}

}

else if (choice == 2)

{

printf("Enter the array:\n");

fflush(stdin);

gets(arrchar);

m=strlen(arrchar);

sortAlphabeticalArray(arrchar, m);

printf("The elements of the character array sorted in ascending order are as follows\n");

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

{

printf("%c",arrchar[i]);

}

}

else

{

printf("Enter one of the given option otherwise the program will not work.\nNow we will have to re-run the program and this time Enter a valid option.");

}

}

**Question 02:**

#include <stdio.h>

#include <string.h>

void sortNumericArray(int arrnum[], int n)

{ printf(“This problem solved by recursion this time\n”);

int i = 0, temp;

if (n > 0)

{

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

{

if (arrnum[i] > arrnum[i + 1])

{

temp = arrnum[i];

arrnum[i] = arrnum[i + 1];

arrnum[i + 1] = temp;

}

}

sortNumericArray(arrnum, n - 1);

}

else

{

return;

}

}

void sortAlphabeticalArray(char arrchar[], int m)

{

int i = 0;

char temp;

if (m >= 0)

{

for (i = 0; i < m-1; i++)

{

if (arrchar[i] > arrchar[i + 1])

{

temp = arrchar[i];

arrchar[i] = arrchar[i + 1];

arrchar[i + 1] = temp;

}

}sortAlphabeticalArray(arrchar,m-1);

}

else

{

return;

}

}

int main()

{

int n, m, choice, i;

char arrchar[1000];

printf("Enter:\n1 for a numeric array\n2 for a character array\n");

scanf("%d", &choice);

if (choice == 1)

{

printf("How many elements do you want in your numeric array?\n");

scanf("%d", &n);

int arrnum[n];

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

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

{

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

}

printf("The elements of the numeric array sorted in ascending order are as follows\n");

sortNumericArray(arrnum, n);

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

{

printf("%d\n", arrnum[i]);

}

}

else if (choice == 2)

{

printf("Enter the array:\n");

fflush(stdin);

gets(arrchar);

m = strlen(arrchar);

sortAlphabeticalArray(arrchar, m);

printf("The elements of the character array sorted in ascending order are as follows\n");

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

{

printf("%c", arrchar[i]);

}

}

else

{

printf("Enter one of the given option otherwise the program will not work.\nNow we will have to re-run the program and this time Enter a valid option.");

}

}

**Question 03:**

**Question 03(By value)**

**1)strcat:**

#include<stdio.h>

#include<string.h>

int my\_strcat(char arr1[],char arr2[]){

int i,j,k;

char res[1000];

i=strlen(arr1);

j=strlen(arr2);

for(k=0;k<=j;k++){

arr1[i]=arr2[k];

i++;

}

printf("Concatinated String\n");

puts(arr1);

}

int main()

{

char arr1[1000],arr2[1000];

printf("Enter the first string\n");

gets(arr1);

printf("Enter the second string\n");

gets(arr2);

printf("After concatenation\n");

my\_strcat(arr1,arr2);

}

**2)Strcopy:**

#include<stdio.h>

#include<string.h>

int func\_stringlength(char arr[])

{

int i=0;

while(arr[i]!='\0')

{

i++;

}

return i;

}

void stringcopy(char arr1[])

{

char arr2[1000];

int n=func\_stringlength(arr1),i;

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

{

arr2[i]=arr1[i];

}

printf("The string which is the copy of the other string is as follows:\n");

puts(arr2);

}

int main()

{

char arr1[1000];

printf("Enter the array you want the copy of\n");

gets(arr1);

stringcopy(arr1);

}

**3)Strlen:**

#include<stdio.h>

int func\_stringlength(char arr[])

{

int i=0;

while(arr[i]!='\0')

{

i++;

}

return i;

}

int main()

{

char arr[1000];

printf("Enter the string below\n");

gets(arr);

printf("The length of the string is %d",func\_stringlength(arr));

}

**4)Strncpy:**

#include<stdio.h>

int func\_stringlength(char arr[])

{

int i=0;

while(arr[i]!='\0')

{

i++;

}

return i;

}

char strncpy(char arr1[],char arr2[],int n)

{

int i;

//n=func\_stringlength(arr2);

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

{

arr1[i]=arr2[i];

}

printf("After putting the given characters of string 2 in string 1\nString 1 becomes\n");

puts(arr1);

}

int main()

{

char arr1[1000],arr2[1000];

int n;

printf("Enter the first string\n");

gets(arr1);

printf("Enter the second string\n");

gets(arr2);

printf("Enter the number of character you want to copy form string 2 to string 1\n");

scanf("%d",&n);

strncpy(arr1,arr2,n);

}

**Question 03(By Ref)**

1. **Strcat**

#include<stdio.h>

#include<string.h>

int my\_strcat(char \*arr1,char \*arr2){

int i,j,k;

char res[1000];

i=strlen(arr1);

j=strlen(arr2);

for(k=0;k<=j;k++){

arr1[i]=arr2[k];

i++;

}

printf("Concatinated String\n");

puts(arr1);

}

int main()

{

char arr1[1000],arr2[1000];

printf("Enter the first string\n");

gets(arr1);

printf("Enter the second string\n");

gets(arr2);

printf("After concatenation\n");

char \*ptr1=&arr1[0],\*ptr2=&arr2[0];

my\_strcat(ptr1,ptr2);

}

1. **Strcopy**

#include<stdio.h>

#include<string.h>

int func\_stringlength(char arr[])

{

int i=0;

while(arr[i]!='\0')

{

i++;

}

return i;

}

void stringcopy(char \*arr1)

{

char arr2[1000];

int n=func\_stringlength(arr1),i;

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

{

arr2[i]=arr1[i];

}

printf("The string which is the copy of the other string is as follows:\n");

puts(arr2);

}

int main()

{

char arr1[1000];

printf("Enter the array you want the copy of\n");

gets(arr1);

char \*ptr=&arr1[0];

stringcopy(ptr);

}

1. **Strlen**

#include<stdio.h>

int func\_stringlength(char \*arr)

{

int i=0;

while(arr[i]!='\0')

{

i++;

}

return i;

}

int main()

{

char arr[1000];

printf("Enter the string below\n");

gets(arr);

char \*ptr=&arr[0];

printf("The length of the string is %d",func\_stringlength(ptr));

}

1. **Strncpy**

#include<stdio.h>

char my\_strncpy(char \*arr1,char \*arr2,int \*n)

{

int i;

for ( i = 0; i <\*n ; i++)

{

arr1[i]=arr2[i];

}

printf("After putting the given characters of string 2 in string 1\nString 1 becomes\n");

puts(arr1);

}

int main()

{

char arr1[1000],arr2[1000];

int n;

printf("Enter the first string\n");

gets(arr1);

printf("Enter the second string\n");

gets(arr2);

printf("Enter the number of character you want to copy form string 2 to string 1\n");

scanf("%d",&n);

char \*ptr1=&arr1[0],\*ptr2=&arr2[0];

int \*ptr3=&n;

my\_strncpy(ptr1,ptr2,ptr3);

}

**Question 04:**

#include <stdio.h>

int pivotpositioner(int arr[], int low, int high)

{

int pivot = arr[low], i = low + 1, temp;

int j = high;

do

{

while (arr[i] <= pivot)

{

i++;

}

while (arr[j] > pivot)

{

j--;

}

if (i < j)

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

} while (i < j);

if (arr[j] < pivot)

{

temp = arr[low];

arr[low] = arr[j];

arr[j] = temp;

}

return j;

}

void quicksort(int arr[], int start, int end)

{

int pivotposition;

if (start < end)

{

pivotposition = pivotpositioner(arr, start, end);

quicksort(arr, start, pivotposition - 1);

quicksort(arr, pivotposition + 1, end);

}

}

int main()

{

int n,i;

printf("How many values do you want in your array?\n");

scanf("%d", &n); int arr[n];

printf("Enter the values of in the array\n");

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

{

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

}

printf("Array before quick sorting\n");

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

{

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

}

quicksort(arr, 0, n - 1);

printf("\nArray after quick sorting\n");

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

{

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

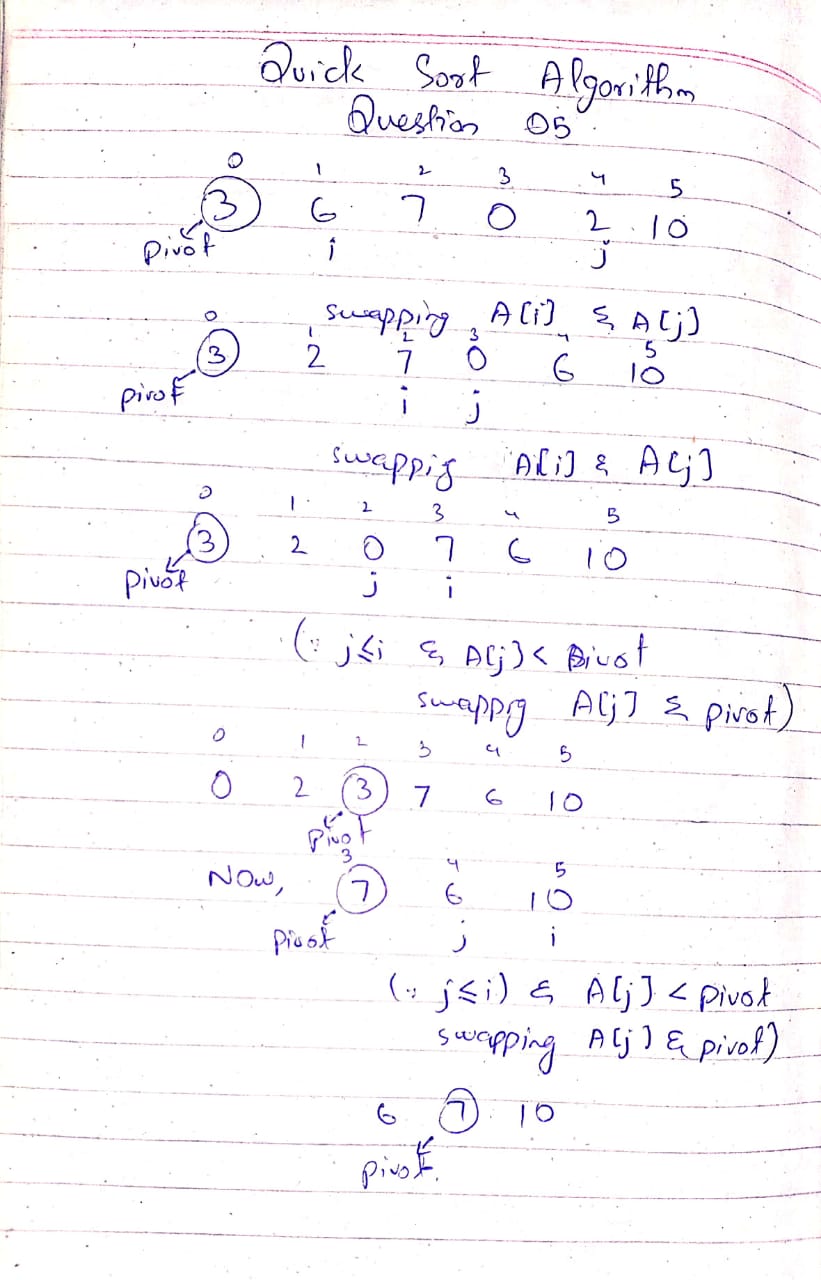
}

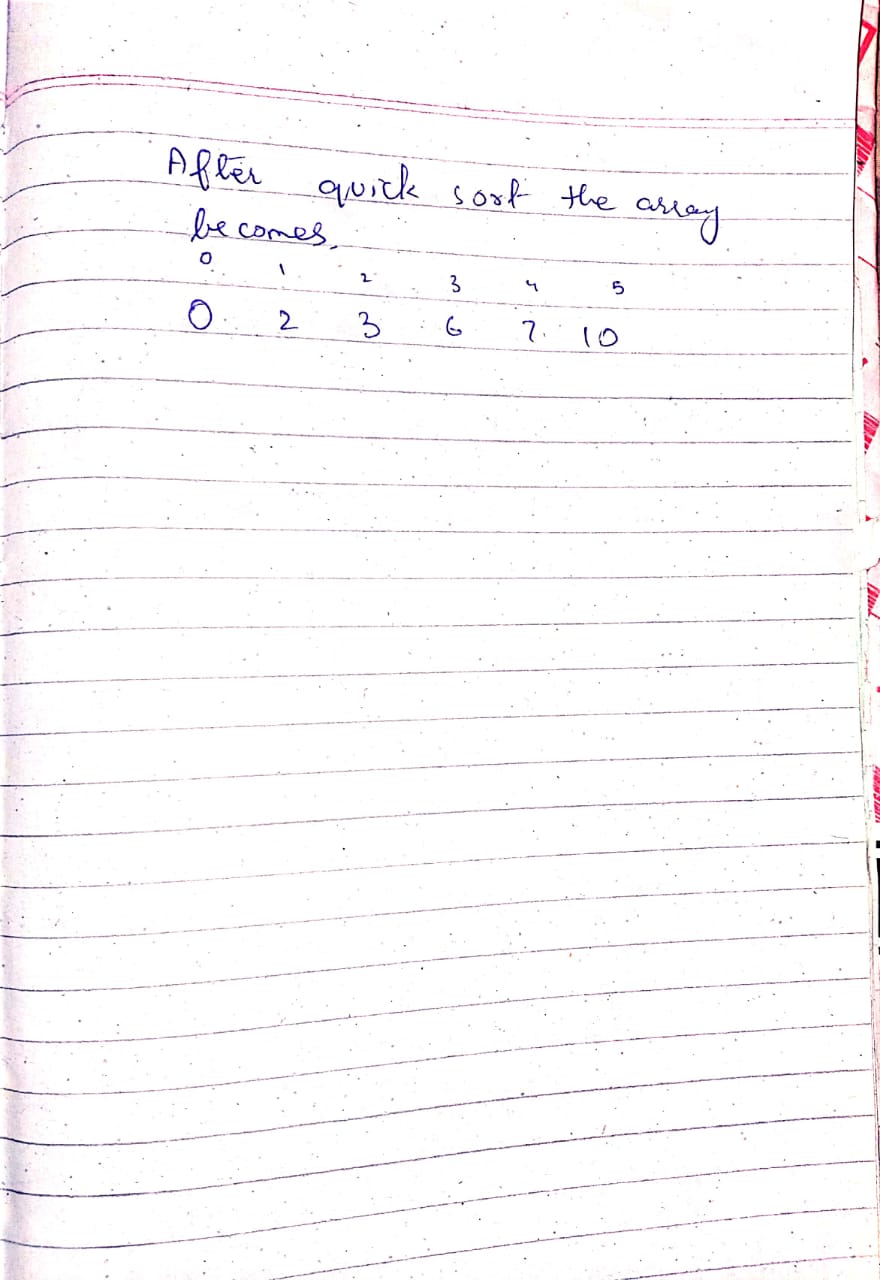
}

**Question 05:**

**Importance of recursion in the given problem:**

Each iteration of the algorithms returns the final position of the pivot. Hence the Algorithm primarily depends upon recursion, through which we determine the final position of the pivot and ultimately arrange all the elements of the given array.

****

****

**Question 06:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

int BS(int a[], int key, int begin, int end)

{

int mid = (begin + end) / 2;

if (a[mid] == key)

return 1;

else

{

if (end <= begin)

return 0;

else

{

if (a[mid] > key)

return BS(a, key, begin, mid - 1);

else

return BS(a, key, mid + 1, end);

}

}

}

int main()

{

int begin, end, key, i, j, temp,n;

printf("How many elements in the array?\n");

scanf("%d",&n);

int arr[n];

printf("Enter the elements below\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])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("Enter the key you want to find\n");

scanf("%d", &key);

begin = arr[0];

end = arr[19];

if (BS(arr, key, begin, end) == 1)

printf("found\n");

else

{

printf("not found\n");

}

}

**Question 07**

Stack:

An everyday example of stack will be a bunch of books placed upon one another. The which is placed first resides at the bottom of the stack, while the book which is placed in the last resides at the top of the stack

Queue:

An everyday example of queue will the line of vehicle on a petrol pump. The vehicle which goes first in the queue get out first while the vehicle which gets in the queue last get out in the last.

**Question 08:**

**Question 08 Part\_1(Inverted right triangle):**

#include<stdio.h>

void Invertedrighttriangle(int n)

{

int i=0;

if(n==1)

printf("\*\n");

else

{

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

{

printf("\*");

}

printf("\n");

n--;

Invertedrighttriangle(n);

}

}

int main()

{

int n;

printf("How many rows do you want in your inverted right triangle?\n");

scanf("%d",&n);

Invertedrighttriangle(n);

}

**Question 08 Part\_2(Rectangle):**

#include<stdio.h>

void rectangle(int rows,int columns)

{ int i;

int static k=1;

if(rows>=k)

{

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

{

printf("\* ");

}

printf("\n");

rows--;

rectangle(rows,columns);

}

else

{

return;

}

}

int main()

{

int rows, columns;

printf("How many rows and columns do you want in your rectangle?\nEnter them below\n");

scanf("%d", &rows);

scanf("%d", &columns);

rectangle(rows,columns);

}