CS266 ASSIGNMENT 1

NAME:

ARCHIT AGRAWAL

ROLL NO. :

202051213

SECTION:

2

**Based on Pointers**

1. Write a program to display address of an integer variable, character variable and string variable using pointer.

***Code***

#include<stdio.h>

void main(){

    //Write a program to display address of an integer variable, character variable and string variable using pointers

    int i = 5;

    char c = 'A';

    char str[] = "Archit";

    int \*addressInt = &i;

    char \*addressChar = &c;

    char \*addressStr = str; //since sting name 'str' is the address of 0th index of string

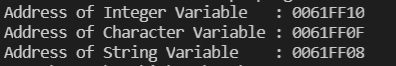
    printf("Address of Integer Variable   : %p\n", addressInt);

    printf("Address of Character Variable : %p\n", addressChar);

    printf("Address of String Variable    : %p\n", addressStr);

}

***Output***



1. Write a program in C to add two numbers using pointers.

***Code***

#include<stdio.h>

int addUsingPointers(int \*a, int \*b){

    return \*a + \*b;

}

void main(){

    int i;

    int j;

    int \*addrI = &i;

    int \*addrJ = &j;

    printf("Enter two numbers : ");

    scanf("%d", addrI);

    scanf("%d", addrJ);

    printf("i = %d, j = %d, i + j = %d", \*addrI, \*addrJ, addUsingPointers(addrI, addrJ));

}

***Output***









1. Write a program in C to find the maximum number between two numbers using a pointer.

***Code***

#include<stdio.h>

int max(int \*a, int \*b){

    return \*a > \*b ? \*a : \*b;

}

void main(){

    int i;

    int j;

    int \*addrI = &i;

    int \*addrJ = &j;

    printf("Enter two numbers : ");

    scanf("%d", addrI);

    scanf("%d", addrJ);

    printf("i = %d, j = %d, maximum of i and j = %d", \*addrI, \*addrJ, max(addrI, addrJ));

}

***Output***









1. Write a program in C to store n elements in an array and print the elements using pointer.

***Code***

#include<Stdio.h>

void main(){

    int n;

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

    scanf("%d", &n);

    int arr[n];

    int \*p = arr;

    printf("Enter the array elements : ");

    for(int i = 0; i < n; i++){

        scanf("%d", p + i);

    }

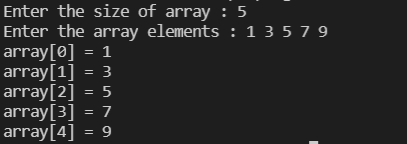
    for(int i = 0; i < n; i++){

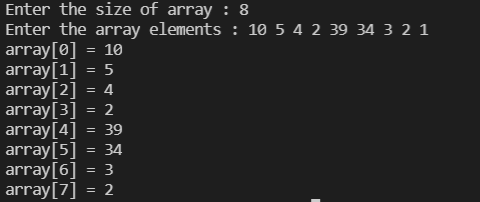
        printf("array[%d] = %d\n", i, \*(p + i));

    }

}

***Output***





1. Write a program in C to Calculate the length of the string using a pointer.

***Code***

#include<Stdio.h>

int stringLength(char str[]){

    int i = 0;

    while(\*(str + i) != '\0'){

        i++;

    }

    return i - 1;

}

void main(){

    char str[1000];

    printf("Enter a string : ");

    fgets(str, sizeof str, stdin);

    printf("%s", str);

    printf("Length of string is %d", stringLength(str));

}

***Output***









1. Write a program in C to sort an array using Pointer.

***Code***

#include<stdio.h>

void bubbleSort(int n, int\* ptr){

    int i, j;

    for (i = 0; i < n; i++){

        for (j = i + 1; j < n; j++){

            if (\*(ptr + j) < \*(ptr + i)){

                int temp = \*(ptr + i);

                \*(ptr + i) = \*(ptr + j);

                \*(ptr + j) = temp;

            }

        }

    }

}

void main(){

    int n;

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

    scanf("%d", &n);

    int arr[n];

    int \*p = arr;

    printf("Enter the array elements : ");

    for(int i = 0; i < n; i++){

        scanf("%d", p + i);

    }

    bubbleSort(n, arr);

    printf("Sorted Array : ");

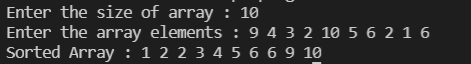
    for(int i = 0; i < n; i++){

        printf("%d ", \*(p + i));

    }

}

***Output***



**Based on File Handling**

1. Write a program in C to create and store information in a text file.

***Code***

#include<stdio.h>

#include<stdlib.h>

void main(){

    FILE \*fptr;

    char fname[] = "question1.txt";

    fptr = fopen(fname, "w");

    if(fptr == NULL){

        printf("File not created");

        exit(1);

    } else {

        printf("File created successfully");

    }

    printf("\nEnter a string to write in the created 'question1.txt' file : ");

    char str[500];

    fgets(str, sizeof str, stdin);

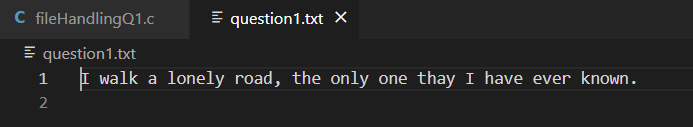
    fputs(str, fptr);

    fclose(fptr);

}

***Output***





1. Write a program in C to count a number of words and characters in a file.

***Code***

#include<stdio.h>

#include<stdlib.h>

void main(){

    FILE \*fptr;

    char fname[20];

    printf("Input the file name to be opened : ");

    scanf("%s", fname);

    fptr = fopen(fname, "r");

    int words = 1;

    int characters = 1;

    if(fptr == NULL){

        printf("Error! File not opened.");

    } else {

        char ch = fgetc(fptr);

        printf("The content in the file %s is:\n\n", fname);

        while(ch!=EOF) {

            printf("%c",ch);

            if(ch==' '||ch=='\n') words++;

            else characters++;

            ch = fgetc(fptr);

        }

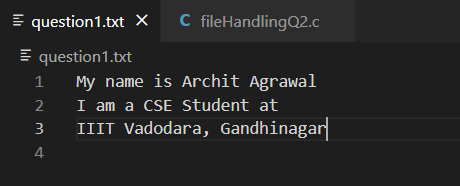
    }

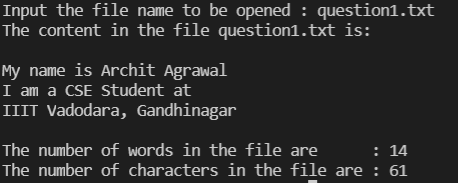
    printf("\nThe number of words in the file are      : %d", words - 2);

    printf("\nThe number of characters in the file are : %d", characters - 1);

}

***Output***





1. Write a program in C to merge two files and write it in a new file.

***Code***

#include<stdio.h>

#include<stdlib.h>

void main(){

    FILE \*file1, \*file2, \*fileNew;

    char fname1[20], fname2[20], fnameNew[20];

    printf("Input the name of first file : ");

    scanf("%s", fname1);

    printf("Input the name of second file : ");

    scanf("%s", fname2);

    file1 = fopen(fname1, "r");

    file2 = fopen(fname2, "r");

    if(file1 == NULL || file2 == NULL){

        printf("Error occured");

        exit(1);

    }

    printf("Enter the name of the new file (in which both files will be merged) : ");

    scanf("%s", fnameNew);

    fileNew = fopen(fnameNew, "w");

    if(fileNew == NULL){

        printf("File not created or error in opening.");

        exit(1);

    } else {

        printf("File %s created successfully.\n", fnameNew);

    }

    //reading the contents from file1 and writing it in fileNew

    char ch;

    while((ch = fgetc(file1)) != EOF){

        fputc(ch, fileNew);

    }

    //reading the contents from file2 and writing it in fileNew

    while((ch = fgetc(file2)) != EOF){

        fputc(ch, fileNew);

    }

    printf("Files %s and %s merged successfully to file %s.", fname1, fname2, fnameNew);

}

***Output***

