Standard Coding Questions (Basic to Advanced Level):

**Basic Programs:**

# Decision Making and Branching constructs

* Write a C program for the following scenario: A telecom service provider offers recharge discount of Rs. 100 for using GPay wallet, Rs. 50 for using Phone Pe wallet and Rs. 200 for using Axis Bank credit card only if in all cases the recharge

amount is more than Rs. 499. Program must accept the Recharge amount and mode of payment. The net amount to be paid must be the output of the program.

Solution:-

* Suppose the postal rates for mailing letters are as follows Rs. 0.50 per 10 grams for the first 50 grams

Rs. 0.40 per 10 grams for the next 100 grams

Rs. 0.25 per 10 grams for the next 250 grams and

Rs. 0.25 per kilograms for the letters weighing more than 400grams

Write a C program that prompts for weight of a letter and prints the postage to be paid

Solution:-

* Write a C program that checks if a given password is strong, weak or moderate as per the rules given below.
  1. If it contains a combination of digits, alphabets and special characters, then the password is strong.
  2. If it contains a combination of only digits and alphabets, then the password is moderate.
  3. If it contains only alphabets, then the password is weak.

Solution:-

#include <stdio.h>

void printStrongNess(string& input)

{

int n = input.length();

bool hasLower = false, hasUpper = false;

bool hasDigit = false, specialChar = false;

string normalChars = "abcdefghijklmnopqrstu"

"vwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 ";

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

if (islower(input[i]))

hasLower = true;

if (isupper(input[i]))

hasUpper = true;

if (isdigit(input[i]))

hasDigit = true;

size\_t special = input.find\_first\_not\_of(normalChars);

if (special != string::npos)

specialChar = true;

}

cout << "Strength of password:-";

if (hasLower && hasUpper && hasDigit &&

specialChar && (n >= 8))

cout << "Strong" << endl;

else if ((hasLower || hasUpper) &&

specialChar && (n >= 6))

cout << "Moderate" << endl;

else

cout << "Weak" << endl;

}

int main()

{

string input = "GeeksforGeeks!@12";

printStrongNess(input);

return 0;

}

* A commercial bank accepts fixed deposit from the public. The rates of interest vary depending on the number of days for which the deposits are made. Following are the rates.

|  |  |
| --- | --- |
| 30 days to 60 days | 6% |
| 61 days to 90 days | 6.5% |
| 91 days to 180 days | 7% |
| 181 days to 1 year | 8% |
| More than 1 year | 8.5% |

For senior citizen, 1% additional interest is offered for deposits more than 50,000. Write a program to accept the amount, number of days for which the amount is deposited and the age of the depositor, calculate the amount to be given to the customer after the completion of term of deposit.

Solution:-

* Write a C program to add two complex numbers

Solution:-

#include <stdio.h>

struct complex

{

int real, img;

};

int main()

{

struct complex a, b, c;

printf("Enter a and b where a + ib is the first complex number.\n");

scanf("%d%d", &a.real, &a.img);

printf("Enter c and d where c + id is the second complex number.\n");

scanf("%d%d", &b.real, &b.img);

c.real = a.real + b.real;

c.img = a.img + b.img;

printf("Sum of the complex numbers: (%d) + (%di)\n", c.real, c.img);

return 0;

}

* Write a C program to read three sides of triangle. Check whether these sides make triangle. If they make then identify whether it is equilateral or isosceles or scalene.

Solution:-

#include<stdio.h>

int main(){

int side1, side2, side3;

printf("Enter sides of triangle:");

scanf("%d%d%d",&side1,&side2,&side3);

if(side1 == side2 && side2 == side3)

printf("The Given Triangle is equilateral\n");

else if(side1 == side2 || side2 == side3 || side3 == side1)

printf("The given Triangle is isosceles\n");

else

printf("The given Triangle is scalene\n");

return 0;

}

* A Student has scored marks in three subjects. Write a C program to find out the highest, second highest and third highest marks among the three subjects. Also find the average of first two highest marks.

Solution:-

#include <stdio.h>

int main ()

{

float mark[10] = {45.6, 78.4, 65.9, 58.3, 82.1, 44.5, 61.8, 53.6, 49.2, 37.7};

int i;

float sum = 0, average, highest = 0;

clrscr();

for (i = 0; i < 10; i++)

{

sum += mark[i];

if (mark[i] > highest)

highest = mark[i];

}

average = sum / 10.0;

printf("The Average Mark is %5.2f \n", average);

printf("The Highest Mark is %5.2f \n", highest);

getch();

return 0;

}

# Decision Making and Looping Statements

* Write a C program that accepts 60 mobile numbers iteratively. On accepting one mobile number the program must check whether the same ends with an even number or odd number. If even, then the contact belongs to Batch A else the contact belongs to Batch B. The program must output the Number of students in Batch A and Batch B after all the 60 mobile numbers are read.

Solution:-

* Suppose an amount is deposited in a commercial Bank , which pays compound interest at the rate of r% annually for n years . Write a C program that prints the amount in account after each year.

Solution:-

#include <stdio.h>

#include <math.h>

int main()

{

float principle, rate, time, CI;

printf("Enter principle (amount): ");

scanf("%f", &principle);

printf("Enter time: ");

scanf("%f", &time);

printf("Enter rate: ");

scanf("%f", &rate);

CI = principle\* (pow((1 + rate / 100), time))

printf("Compound Interest = %f", CI);

return 0;

}

* Write a C program to create a simple gamming application as per the scenario. The admin is asked to select a number between 1 to 10. The user is asked to enter any number. If the number matches the number chosen by the admin then the user wins. The user is given maximum of 3 chances.

Solution:-

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void guess(int N)

{

int number, guess, numberofguess = 0;

srand(time(NULL));

number = rand() % N;

printf("Guess a number between"

" 1 and %d\n",

N);

do {

if (numberofguess > 9) {

printf("\nYou Loose!\n");

break;

}

scanf("%d", &guess);

if (guess > number)

{

printf("Lower number "

"please!\n");

numberofguess++;

}

else if (number > guess)

{

printf("Higher number"

" please!\n");

numberofguess++;

}

else

printf("You guessed the"

" number in %d "

"attempts!\n",

numberofguess);

} while (guess != number);

}

int main()

{

int N = 100;

guess(N);

return 0;}

* Write a program to generate following pattern

1

2 1 2

3 2 1 2 3

4 3 2 1 2 3 4

Solution:-

#include<stdio.h>

#include<conio.h>

int main()

{

int i, s, n, j = 0, c = 0, c1 = 0;

printf("Enter the number of rows: ");

scanf("%d",&n);

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

{

for(s = 1; s <= n-i; ++s)

{

printf(" ");

++c;

}

while(j != 2 \* i - 1)

{

if (c <= n - 1)

{

printf("%d ", i + j);

++c;

}

else

{

++c1;

printf("%d ", (i + j - 2 \* c1));

}

++j;

}

c1 = c = j = 0;

printf("\n");}return 0 }

* Write a C program to find Strong Numbers within a range of numbers.

Solution:-

#include <stdio.h>

int main()

{

int n;

int sum=0;

printf("Enter a number");

scanf("%d",&n);

int k=n;

int r;

while(k!=0)

{

r=k%10;

int f=fact(r);

k=k/10;

sum=sum+f;

}

if(sum==n)

{

printf("\nNumber is a strong");

}

else

{

printf("\nNumber is not a strong");

}

return 0;

}

int fact(int r)

{

int mul=1;

for(int i=1;i<=r;i++)

{

mul=mul\*i; } return mul; }

* Write a C program to print following pattern.

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

Solution:-

#include <stdio.h>

void main()

{

int i,j,n;

printf("Input number of rows : ");

scanf("%d",&n);

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

{

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

printf(" ");

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

printf("%d",j);

for(j=i-1;j>=1;j--)

printf("%d",j);

printf("\n");

}

}

* Write a C program to to calculate the average hight and weight of the entire class and categorise into

i) Avg\_Height <5feet. ii) 5< Avg\_Height<6 iii) Avg\_Weight<50kg

iv) 50< Avg\_Weight<75kg

solution:-

#include <stdio.h>

void main()

{

float height;

printf("Enter the Height (in centimetres) \n");

scanf("%f", &height);

if (height < 150.0)

printf("Dwarf \n");

else if ((height >= 150.0) && (height <= 165.0))

printf(" Average Height \n");

else if ((height > 165.0) && (height <= 195.0))

printf("Taller \n");

else

printf("Abnormal height \n");

}

# Single dimension Array

* Write a C program to read a mobile number. Create a single dimension array in such a way that the index location of the array that matches the digit present in the mobile number must be filled with the frequency of that digit in the mobile

number. Example if the mobile number is 9945690021 then the array location with index 9 must be filled with 3 as 9 appears 3 times in the mobile number.

Solution:-

* Write a C program that accepts an array , interchanges first element with last element , second element with second last element & so on , finally prints the new array.

Solution:-

#include <stdio.h>

void Array\_Swap(int \*array , int n)

{

int i=0,temp=0;

for(i=0 ; i<n/2 ; i++)

{

temp = array[i];

array[i] = array[n-i-1];

array[n-i-1] = temp;

}

}

int main()

{

int array\_1[30] = {0};

int i=0 ,n=0;

printf("\nEnter the number of elements for the array : ");

scanf("%d",&n);

printf("\nEnter the elements for array\_1..\n");

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

{

printf("array\_1[%d] : ",i);

scanf("%d",&array\_1[i]);

}

Array\_Swap(array\_1 , n);

printf("\nThe array after swap is..\n");

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

{

printf("\narray\_1[%d] : %d",i,array\_1[i]);

}

return 0;

}

* Write a C program to cyclically display the elements of an array for any number of digits.

Sample input: 1 2 3

Sample output: 123 231 312.

Solution:

#include <stdio.h>

int countdigits(int N)

{

int count = 0;

while (N) {

count++;

N = N / 10;

}

return count;

}

void cyclic(int N)

{

int num = N;

int n = countdigits(N);

while (1) {

print("%d",num)

int rem = num % 10;

int div = num / 10;

num = (pow(10, n - 1)) \* rem + div;

if (num == N)

break;

}

}

int main()

{

int N = 5674;

cyclic(N);

return 0;}

* Write a program to find intersection of two arrays a and b with size m and n respectively.

Example:

int a[10]={1,2,4,6,7,8,11,12,13,5}

intb[5]={8,12,56,13}

Output:

c[10]={8,12,13}

Solution:-

* Write a C program to count the frequencies of elements of a given integer array.

Solution:-

#include <stdio.h>

int main()

{

int arr[] = {1, 2, 8, 3, 2, 2, 2, 5, 1};

int length = sizeof(arr)/sizeof(arr[0]);

int fr[length];

int visited = -1;

for(int i = 0; i < length; i++){

int count = 1;

for(int j = i+1; j < length; j++){

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

count++;

fr[j] = visited;

} }

if(fr[i] != visited)

fr[i] = count;

}

printf("---------------------\n");

printf(" Element | Frequency\n");

printf("---------------------\n");

for(int i = 0; i < length; i++){

if(fr[i] != visited){

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

printf(" | ");

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

}

}

printf("---------------------\n");

return 0;

}

* Write a C program to read n numbers in 1-D array and sum of any consecutive elements present in array. Print the sub-array making sum along with start and end index.

Solution:

#include<stdio.h>

int sum(int arr[], int n)

{

int sum = 0;

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

sum += arr[i];

return sum;

}

int main()

{

int arr[] = {12, 3, 4, 15};

int n = sizeof(arr) / sizeof(arr[0]);

printf("Sum of given array is %d", sum(arr, n));

return 0;

}

* Write a C program to count the number of vowels in a given words.

Solution:-

#include <stdio.h>

int main()

{

int c = 0, count = 0;

char s[1000];

printf("Input a string\n");

gets(s);

while (s[c] != '\0') {

if (s[c] == 'a' || s[c] == 'A' || s[c] == 'e' || s[c] == 'E' || s[c] == 'i' || s[c] == 'I' ||

s[c] =='o' || s[c]=='O' || s[c] == 'u' || s[c] == 'U')

count++;

c++;

}

printf("Number of vowels in the string: %d", count);

return 0;

**Double dimension Array**:

* Write a C program to read Physics, Chemistry and Maths marks of 20 students. The total marks secured must be calculated. The maximum marks per student is

300. The PCM percentage also must be calculated. All this data must be stored in an array. The output is a neat display of data shown as under:

Physics Chemistry Maths Marks Secured Total Marks Percentage

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 52 |  | 72 |  | 80 |  | 204 |  | 300 | 68.00 |
| 40 |  | 30 |  | 50 |  | 120 |  | 300 | 40.00 |

Given a matrix A of order mxn , write a C program to sort even rows in ascending order and odd rows in descending order.

Solution:-

#include <stdio.h>

int main()

{

int phy, chem, bio, math, comp;

float per;

printf("Enter PCM subjects marks: ");

scanf("%d%d%d%d%d", &phy, &chem, &math);

per = (phy + chem + math ) / 3.0;

printf("Percentage = %.2f\n", per);

if(per >= 90)

{

printf("Grade A");

}

else if(per >= 80)

{

printf("Grade B");

}

else if(per >= 70)

{

printf("Grade C");

}

else if(per >= 60)

{

printf("Grade D");

}

else if(per >= 40)

{

printf("Grade E");

}

else

{

printf("Grade F");

}

return 0;

}

* Write a C program to find the largest and smallest words based on the

number of characters in a two dimensional character array without using library functions.

Solution:-

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void main()

{

char str[100], word[20], mx[20], mn[20], c;

int i = 0, j = 0, flg = 0;

printf("\n\nFind the largest and smallest word in a string :\n");

printf("-----------------------------------------------------\n");

printf("Input the string : ");

i = 0;

do

{

fflush(stdin);

c = getchar();

str[i++] = c;

} while (c != '\n');

str[i - 1] = '\0';

for (i = 0; i < strlen(str); i++)

{

while (i < strlen(str) && !isspace(str[i]) && isalnum(str[i]))

{

word[j++] = str[i++];

}

if (j != 0)

{

word[j] = '\0';

if (!flg)

{

flg = !flg;

strcpy(mx, word);

strcpy(mn, word);

}

if (strlen(word) > strlen(mx))

{

strcpy(mx, word);

}

if (strlen(word) < strlen(mn))

{

strcpy(mn, word);

}

j = 0;

}

}

printf("The largest word is '%s' \nand the smallest word is '%s' \nin the string : '%s'.\n", mx, mn, str);

}

* Annual examination results of 100 students is stored as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Roll No. | Subject 1 | Subject 2 | Subject 3 |
| :  :  : | :  :  : | :  :  : | :  :  : |

Write a program to read the data and determine the following: (a) Total marks obtained by each student

(b) The highest marks in each subject and roll number of the student

who secured it.

c) The student who has secured highest marks(class topper)

Solution:-

#include<stdio.h>

struct student

{

int sub1;

int sub2;

int sub3;

};

void main()

{

struct student s[10];

int i,total=0;

clrscr();

for(i=0;i<=2;i++)

{

printf("\nEnter Marks in Three Subjects = ");

scanf("%d%d%d",& s[i].sub1,&s[i].sub2,&s[i].sub3);

total=s[i].sub1+s[i].sub2+s[i].sub3;

printf("\nTotal marks of s[%d] Student= %d",i,total);

}

getch();

}

* The given are two one dimensional arrays A and B which are sorted in ascending order. Write a C program to merge them into a single sorted array C that contains every element from array A and B, in ascending order.

Solution:-

int main()

{

int a[10],b[10],c[20],n1,n2,i,j,temp,k=0;

clrscr();

printf(” Enter the no. of element for 1st array : “);

scanf(“%d”,&n1);

for(i=0;i<n1;i++,k++)

{

printf(” Enter element [%d] : “,i+1);

scanf(“%d”,&a[i]);

c[k]=a[i];

}

for(i=0;i<n1;i++)

{

for(j=i+1;j<n1;j++)

{

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

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf(“\n After sorting 1st array : “);

for(i=0;i<n1;i++)

{

printf(“\n Element [%d] = %d”,i+1,a[i]);

}

printf(“\n\n Enter the no. of element for 2nd array : “);

scanf(“%d”,&n2);

for(i=0;i<n2;i++,k++)

{

printf(” Enter element [%d] : “,i+1);

scanf(“%d”,&b[i]);

c[k]=b[i];

}

for(i=0;i<n2;i++)

{

for(j=i+1;j<n2;j++)

{

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

{

temp=b[i];

b[i]=b[j];

b[j]=temp;

}

}

}

printf(“\n After sorting 2nd array : “);

for(i=0;i<n2;i++)

{

printf(“\n Element [%d] = %d”,i+1,b[i]);

}

for(i=0;i<n1+n2;i++)

{

for(j=i+1;j<n1+n2;j++)

{

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

{

temp=c[i];

c[i]=c[j];

c[j]=temp;

}

}

}

printf(“\n\n\n After combined and sorted both array :- “);

for(i=0;i<n1+n2;i++)

{

printf(“\n Element [%d] = %d”,i+1,c[i]);

}

getch();

}

* Write a C program to store product code and price in 2-D array, calculate the total price for multiple product codes entered from array along with quantity.

Solution:-

#include <stdio.h>

const int CITY = 2;

const int WEEK = 7;

int main()

{

int temperature[CITY][WEEK];

for (int i = 0; i < CITY; ++i)

{

for (int j = 0; j < WEEK; ++j)

{

printf("City %d, Day %d: ", i + 1, j + 1);

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

}

}

printf("\nDisplaying values: \n\n");

for (int i = 0; i < CITY; ++i)

{

for (int j = 0; j < WEEK; ++j)

{

printf("City %d, Day %d = %d\n", i + 1, j + 1, temperature[i][j]);

}

}

return 0;

}

* Write a C program to store the students name and USN using 2 dimensional array and display the list.

Solution:-

#include <stdio.h>

struct student {

char firstName[50];

int roll;

float marks;

} s[5];

int main() {

int i;

printf("Enter information of students:\n");

for (i = 0; i < 5; ++i) {

s[i].roll = i + 1;

printf("\nFor roll number%d,\n", s[i].roll);

printf("Enter first name: ");

scanf("%s", s[i].firstName);

printf("Enter marks: ");

scanf("%f", &s[i].marks);

}

printf("Displaying Information:\n\n");

for (i = 0; i < 5; ++i) {

printf("\nRoll number: %d\n", i + 1);

printf("First name: ");

puts(s[i].firstName);

printf("Marks: %.1f", s[i].marks);

printf("\n");

}

return 0;

}

# String Manipulation:

* Write a C program to sort the characters of a string in ascending order. The string must contain only small case alphabets.

Solution:-

#include <stdio.h>

#include <string.h>

char input[100];

void ascendingOrderString() {

int i, j;

char temp;

int stringLength = strlen(input);

for (i = 0; i < stringLength - 1; i++) {

for (j = i + 1; j < stringLength; j++) {

if (input[i] > input[j]) {

temp = input[i];

input[i] = input[j];

input[j] = temp;

}

}

}

}

int main() {

printf("\n\t Enter the string : ");

fgets(input, 100, stdin);

ascendingOrderString();

puts(input);

return 0;

}

* Write a C program to illustrate input and output of array of strings represented as 2D array of characters. (eg Days in a week or Months in a year)

Solution:-

#include<stdio.h>

int main(){

int disp[2][3];

int i, j;

for(i=0; i<2; i++) {

for(j=0;j<3;j++) {

printf("Enter value for disp[%d][%d]:", i, j);

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

}

}

printf("Two Dimensional array elements:\n");

for(i=0; i<2; i++) {

for(j=0;j<3;j++) {

printf("%d ", disp[i][j]);

if(j==2){

printf("\n");

}

}

}

return 0;

}

* Write a C program to remove 3 or more consecutive characters from a string, repeat until there are no more

Ex: MNHHHHNNM => MNNNM => MM

Solution:-

#include <stdio.h>

int main() {

char str[100];

int i, j, len, len1;

printf("Enter any string: ");

gets(str);

for (len = 0; str[len] != '\0'; len++);

len1 = 0;

for (i = 0; i < (len - len1);) {

if (str[i] == str[i + 1]) {

for (j = i; j < (len - len1); j++)

str[j] = str[j + 1];

len1++;

} else {

i++;

}

}

printf("String after removing characters: %s\n", str);

* Write a program to find number of occurrences of a word in a string.

Solution:-

#include <stdio.h>

#include <string.h>

#define MAX\_SIZE 100

int countOccurrences(char \* str, char \* toSearch);

int main()

{

char str[MAX\_SIZE];

char toSearch[MAX\_SIZE];

int count;

printf("Enter any string: ");

gets(str);

printf("Enter word to search occurrences: ");

gets(toSearch);

count = countOccurrences(str, toSearch);

printf("Total occurrences of '%s': %d", toSearch, count);

return 0;

}

int countOccurrences(char \* str, char \* toSearch)

{

int i, j, found, count;

int stringLen, searchLen;

stringLen = strlen(str);

searchLen = strlen(toSearch);

count = 0;

for(i=0; i <= stringLen-searchLen; i++)

{

found = 1;

for(j=0; j<searchLen; j++)

{

if(str[i + j] != toSearch[j])

{

found = 0;

break;

}

}

if(found == 1)

{

count++;

}

}

return count;

}

* Write a C program which will read a string and rewrite it in the alphabetical order.

For example the word STRING should be written as GINRST.

Solution:-

#include<stdio.h>

int main()

{

char str[100],temp;

int i,j;

clrscr();

printf("Enter the string :");

gets(str);

printf("%s in ascending order is -> ",str);

for(i=0;str[i];i++)

{

for(j=i+1;str[j];j++)

{

if(str[j]<str[i])

{

temp=str[j];

str[j]=str[i];

str[i]=temp;

}

}

}

printf("%s\n",str);

getch();

return 0;

}

* Write a program in C to print all permutations of a given string using pointers.

Solution:-

#include <stdio.h>

#include <string.h>

void changePosition(char \*ch1, char \*ch2)

{

char tmp;

tmp = \*ch1;

\*ch1 = \*ch2;

\*ch2 = tmp;

}

void charPermu(char \*cht, int stno, int endno)

{

int i;

if (stno == endno)

printf("%s ", cht);

else

{

for (i = stno; i <= endno; i++)

{

changePosition((cht+stno), (cht+i));

charPermu(cht, stno+1, endno);

changePosition((cht+stno), (cht+i));

}

}

}

int main()

{

char str[] = "abcd";

printf("\n\n Pointer : Generate permutations of a given string :\n");

printf("--------------------------------------------------------\n");

int n = strlen(str);

printf(" The permutations of the string are : \n");

charPermu(str, 0, n-1);

printf("\n\n");

return 0;

}

* Write a C program to read paragraphs of lines. Count the occurrences of a given

word.

Solution:-

#include <stdio.h>

#include <string.h>

int main()

{

char s[1000],w[1000];

int n,a[1000],i,j,k=0,l,found=0,t=0;

printf("Enter the string : ");

gets(s);

printf("Enter word to be searched: ");

gets(w);

for(i=0;s[i];i++)

{

if(s[i]==' ')

{

a[k++]=i;

}

}

a[k++]=i;

j=0;

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

{

n=a[i]-j;

if(n==strlen(w))

{

t=0;

for(l=0;w[l];l++)

{

if(s[l+j]==w[l])

{

t++;

}

}

if(t==strlen(w))

{

found++;

}

}

j=a[i]+1;

}

printf("word '%s' is occurred count=%d ",w,found);

} 0;

}

* Using String manipulation functions, write a C program to read the name, USN, address and Branch.

Solution:-

#include <stdio.h>

struct student {

char name[50];

int roll;

float marks;

} s;

int main() {

printf("Enter information:\n");

printf("Enter name: ");

fgets(s.name, sizeof(s.name), stdin);

printf("Enter roll number: ");

scanf("%d", &s.roll);

printf("Enter marks: ");

scanf("%f", &s.marks);

printf("Displaying Information:\n");

printf("Name: ");

printf("%s", s.name);

printf("Roll number: %d\n", s.roll);

printf("Marks: %.1f\n", s.marks);

return}

# User Defined Function:

* Write a program that reads 10 mobile numbers and stores them in an array. Each number must be sorted in ascending order using a sort function. Print the resultant array with all numbers.

Solution:-

#include <stdio.h>

int main()

{

int arr[] = {5, 2, 8, 7, 1};

int temp = 0;

int length = sizeof(arr)/sizeof(arr[0]);

printf("Elements of original array: \n");

for (int i = 0; i < length; i++) {

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

}

for (int i = 0; i < length; i++) {

for (int j = i+1; j < length; j++) {

if(arr[i] > arr[j]) {

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("\n");

printf("Elements of array sorted in ascending order: \n");

for (int i = 0; i < length; i++) {

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

}

return 0;

}

* 2. Write a function , say int unit\_digit(int n) that returns the unit digit of a number represented in argument n.

Solution:-

#include <stdio.h>

void printMultiples(int n)

{

int unit\_digit = n % 10;

if (unit\_digit == 0)

unit\_digit = 10;

for (int i = unit\_digit; i <= n; i += unit\_digit)

cout << i << " ";

}

int main()

{

int n = 39;

printMultiples(n);

return 0;

}

* 3. Write a C program which displays the details of a criminal in the police station. The name of the criminal is sent as the parameter to a function which displays the details of the criminal such as his name, his criminal record, place of birth, the term for which he is held in imprisonment and other details. If the name doesn’t match with the existing names then a message should be displayed as “Record not found”.

Solution:-

* 4. Write a program to convert a decimal number to its binary equivalent using function.

Solution:-

#include <stdio.h>

int main(){

int arr[10], num, i, j;

printf("Enter a decimal number: ");

scanf("%d", &num);

for (i = 0; num > 0; i++){

arr[i] = num % 2;

num = num / 2;

}

printf("Equivalent binary number is: ");

for (j = i - 1; j >= 0; j--){

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

}

return 0;

}

* 5. Write a C program to perform matrix addition and subtraction using function in following way

1. read\_matrix(int [ ][ ]) to read matrix
2. operation(int [ ][ ], int) to perform addition / subtraction depending on choice iii. print\_matrix(int [ ][ ]) to print matrix

solution:-

#include<stdio.h>

main(){

int m,n,i,j;

printf("Enter the size of the matrices:\nNo. of rows (m): ");

scanf("%d",&m);

printf("\nNo. of columns(n): ");

scanf("%d",&n);

double a[m][n];

double b[m][n];

double sum[m][n];

printf("\nEnter the elements of matrix A:\n");

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

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

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

}

}

printf("\nEnter the elements of matrix B:\n");

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

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

scanf("%lf",&b[i][j]);

}

}

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

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

sum[i][j]=a[i][j]+b[i][j];

}

}

printf("\nThe sum of the matrices A and B is:\n");

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

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

printf("%lf \t",sum[i][j]);

}

printf("\n");

}}

* Write a C program to demonstrate the use of Calculator using the user defined functions.

Solution:-

#include <stdio.h>

int main()

{

char opt;

int n1, n2;

float res;

printf (" Choose an operator(+, -, \*, /) to perform the operation in C Calculator \n ");

scanf ("%c", &opt);

if (opt == '/' )

{

printf (" You have selected: Division");

}

else if (opt == '\*')

{

printf (" You have selected: Multiplication");

}

else if (opt == '-')

{

printf (" You have selected: Subtraction");

}

else if (opt == '+')

{

printf (" You have selected: Addition");

}

printf (" \n Enter the first number: ");

scanf(" %d", &n1);

printf (" Enter the second number: ");

scanf (" %d", &n2);

switch(opt)

{

case '+':

res = n1 + n2;

printf (" Addition of %d and %d is: %.2f", n1, n2, res);

break;

case '-':

printf (" Subtraction of %d and %d is: %.2f", n1, n2, res);

break;

case '\*':

res = n1 \* n2;

printf (" Multiplication of %d and %d is: %.2f", n1, n2, res);

break;

case '/':

if (n2 == 0)

{

printf (" \n Divisor cannot be zero. Please enter another value ");

scanf ("%d", &n2);

}

res = n1 / n2;

printf (" Division of %d and %d is: %.2f", n1, n2, res);

break;

default:

printf (" Something is wrong!! Please check the options ");

}

return 0; }

# Recursion:

* Write a C program that prints the Sum of first n natural numbers using recursion

Solution:-

#include <stdio.h>

int addNumbers(int n);

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

printf("Sum = %d", addNumbers(num));

return 0;

}

int addNumbers(int n) {

if (n != 0)

return n + addNumbers(n - 1);

else

return n;

}

* Write a C program to find sum of elements of an array named ‘a’ with ‘n’ elements using recursion.

Solution

#include <stdio.h>

#define MAX\_SIZE 100

int sum(int arr[], int start, int len);

int main()

{

int arr[MAX\_SIZE];

int N, i, sumofarray;

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

scanf("%d", &N);

printf("Enter elements in the array: ");

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

{

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

}

sumofarray = sum(arr, 0, N);

printf("Sum of array elements: %d", sumofarray);

return 0;

int sum(int arr[], int start, int len)

{

if(start >= len)

return 0;

return (arr[start] + sum(arr, start + 1, len));

}

* Write a C program to print the sum of first n odd numbers using recursion.

Solution:-

#include<stdio.h>

int SumOdd(int num1, int num2)

{

if(num1>num2)

return 0;

return num1+SumOdd(num1+2,num2);

}

int main()

{

int num1=1,num2;

printf("Enter your Limit:");

scanf("%d",&num2);

printf("Sum of all odd numbers in the given range is: %d",SumOdd(num1,num2));

}

* Write a recursive function to implement binary search.

Solution:-

#include <stdio.h>

int iterativeBinarySearch(int array[], int start\_index, int end\_index, int element){

while (start\_index <= end\_index){

int middle = start\_index + (end\_index- start\_index )/2;

if (array[middle] == element)

return middle;

if (array[middle] < element)

start\_index = middle + 1;

else

end\_index = middle - 1;

}

return -1;

}

int main(void){

int array[] = {1, 4, 7, 9, 16, 56, 70};

int n = 7;

int element = 16;

int found\_index = iterativeBinarySearch(array, 0, n-1, element);

if(found\_index == -1 ) {

printf("Element not found in the array ");

}

else {

printf("Element found at index : %d",found\_index);

} return 0;}

* Write a program in C to check a number is a prime number or not using recursion.

Solution:-

#include <stdio.h>

int primeno(int, int);

int main()

{

int num, check;

printf("Enter a number: ");

scanf("%d", &num);

check = primeno(num, num / 2);

if (check == 1)

{

printf("%d is a prime number\n", num);

}

else

{

printf("%d is not a prime number\n", num);

}

return 0;

}

int primeno(int num, int i)

{

if (i == 1)

{

return 1;

}

else

{

if (num % i == 0)

{

return 0;

}

else

{

return primeno(num, i - 1);

}

}

}

* Write a C program to generate and print first 10 natural numbers along with its sum using recursion

**Solution:-**

**#include <stdio.h>**

**int addNumbers(int n);**

**int main() {**

**int num;**

**printf("Enter a positive integer: ");**

**scanf("%d", &num);**

**printf("Sum = %d", addNumbers(num));**

**return 0;**

**}**

**int addNumbers(int n) {**

**if (n != 0)**

**return n + addNumbers(n - 1);**

**else**

**return n;**

**}**

* Write a C program to find the Fibonacci series and also add all the numbers using recursive function.

Solution:-

#include<stdio.h>

int main()

{

int n1=0,n2=1,n3,i,number;

printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n%d %d",n1,n2);

for(i=2;i<number;++i)

{

n3=n1+n2;

printf(" %d",n3);

n1=n2;

n2=n3;

}

return 0;

}

# Pointers :

* Write a program to read 10 numbers using pointers. Accept a number and search whether that number is present in the list. Use Pointer concept while searching.

Solution:-

#include<stdio.h>

int i,l;

int search(int ,int \*,int);

int main(){

int n,m;

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

scanf("%d",&n);

int a[n];

printf("enter the elements:\n");

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

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

}

printf("enter the element to be searched:");

scanf("%d",&m);

search(n,a,m);

return 0;

}

int search(int n,int \*a,int m){

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

if(m==a[i]){

l=1;

break;

}

}

if(l==1){

printf("%d is present in the array",m);

} else {

printf("%d is not present in the array",m);

}}

* Write a program that dynamically allocates memory for an array of size n , reads elements of array & sorts them in ascending order.

Solution:-

#include <stdio.h>

#include <stdlib.h>

int main()

{

int\* ptr;

int n, i;

printf("Enter number of elements:");

scanf("%d",&n);

printf("Entered number of elements: %d\n", n);

ptr = (int\*)malloc(n \* sizeof(int));

if (ptr == NULL) {

printf("Memory not allocated.\n");

exit(0);

}

else {

printf("Memory successfully allocated using malloc.\n");

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

ptr[i] = i + 1;

}

printf("The elements of the array are: ");

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

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

}

}

return 0;

}

* Write a C program that helps a kid to remember alphabets where in a character is given as an input and a fruit name or a country name is displayed beginning with that character.

Solution:-

#include <stdio.h>

int main() {

char c;

printf("Enter a character: ");

scanf("%c", &c);

if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))

printf("%c is an alphabet.", c);

else

printf("%c is not an alphabet.", c);

return 0;

}

* Write a program using functions and pointers to implement substrptr

(char s1[], char s2[]) with two arguments of character array type containing source string (s1)and substring(s2). Function should return an address of a substring in a string if present else a NULL pointer.

Solution:-

* Write a C program to change the value of integer constant using pointer.

Solution:-

#include <stdio.h>

int main()

{

const int a=10;

int \*p;

p=&a;

printf("Before changing - value of a: %d",a);

\*p=20;

printf("\nAfter changing - value of a: %d",a);

printf("\nWauuuu... value has changed.");

return 0;

}

* Write a C program to swap two numbers using pointers.

Solution:-

#include <stdio.h>

int main()

{

int x, y, \*a, \*b, temp;

printf("Enter the value of x and y\n");

scanf("%d%d", &x, &y);

printf("Before Swapping\nx = %d\ny = %d\n", x, y);

a = &x;

b = &y;

temp = \*b;

\*b = \*a;

\*a = temp;

printf("After Swapping\nx = %d\ny = %d\n", x, y);

return 0;

}

# Structures:

* Write a C program to store 10 student records (Name, USN Number, Mobile Number, Address) and sort them by Name in ascending order. Output the sorted records.

Solution:-

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Student {

char\* name;

int id;

char age;

};

int comparator(const void\* p, const void\* q)

{

return strcmp(((struct Student\*)p)->name,

((struct Student\*)q)->name);

}

int main()

{

int i = 0, n = 5;

struct Student arr[n];

arr[0].id = 1;

arr[0].name = "bd";

arr[0].age = 12;

arr[1].id = 2;

arr[1].name = "ba";

arr[1].age = 10;

arr[2].id = 3;

arr[2].name = "bc";

arr[2].age = 8;

arr[3].id = 4;

arr[3].name = "aaz";

arr[3].age = 9;

arr[4].id = 5;

arr[4].name = "az";

arr[4].age = 10;

printf("Unsorted Student Records:\n");

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

printf("Id = %d, Name = %s, Age = %d \n",

arr[i].id, arr[i].name, arr[i].age);

}

qsort(arr, n, sizeof(struct Student), comparator);

printf("\n\nStudent Records sorted by Name:\n");

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

printf("Id = %d, Name = %s, Age = %d \n",

arr[i].id, arr[i].name, arr[i].age);

}

return 0;

}

* Consider the following declaration

Struct Emp{

int code;

char name[20];

float salary;

};

Given that there are n (<=50) employees in an organization .Write a program using pointer notation to access the data of an Employee that outputs the details of the employee having highest salary.

Solution:-

* Write a C program for generating bill in a shopping mall. The bill should contain the

item description, quantity of items purchased, rate and finally the total amount.

Solution:-

#include <stdio.h>

void main()

{

struct date

{

int day;

int month;

int year;

};

struct details

{

char name[20];

int price;

int code;

int qty;

struct date mfg;

};

struct details item[50];

int n, i;

printf("Enter number of items:");

scanf("%d", &n);

fflush(stdin);

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

{

fflush(stdin);

printf("Item name: \n");

scanf("%s", item[i].name);

fflush(stdin);

printf("Item code: \n");

scanf("%d", &item[i].code);

fflush(stdin);

printf("Quantity: \n");

scanf("%d", &item[i].qty);

fflush(stdin);

printf("price: \n");

scanf("%d", &item[i].price);

fflush(stdin);

printf("Manufacturing date(dd-mm-yyyy): \n");

scanf("%d-%d-%d", &item[i].mfg.day,

&item[i].mfg.month, &item[i].mfg.year);

}

printf(" \*\*\*\*\* INVENTORY \*\*\*\*\* \n");

printf("---------------------------------------------------------

---------\n");

printf("S.N.| NAME | CODE | QUANTITY | PRICE

| MFG.DATE \n");

printf("---------------------------------------------------------

---------\n");

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

printf("%d %-15s %-d %-5d %-5d

%d/%d/%d \n", i + 1, item[i].name, item[i].code, item[i].qty,

item[i].price, item[i].mfg.day, item[i].mfg.month,

item[i].mfg.year);

printf("---------------------------------------------------------

---------\n");

}

* Define structure by name date with members day, month and year of int type. Perform following:
  1. Validate a date
  2. if d1 and d2 are two dates, find whether d1 is earlier than d2 or d1 is later than d2 or both are equal.
  3. find difference between two dates d1 and d2 in terms of number of days.

Solution:-

* Read information (name, roll no and marks) of 10 students using structures and Write a C program to sort them using roll no in ascending order.

Solution:-

#include <stdio.h>

struct student {

char name[50];

int roll;

float marks;

} s;

int main() {

printf("Enter information:\n");

printf("Enter name: ");

fgets(s.name, sizeof(s.name), stdin);

printf("Enter roll number: ");

scanf("%d", &s.roll);

printf("Enter marks: ");

scanf("%f", &s.marks);

printf("Displaying Information:\n");

printf("Name: ");

printf("%s", s.name);

printf("Roll number: %d\n", s.roll);

printf("Marks: %.1f\n", s.marks);

return 0;

}

* Write a C Program to read marks of three subjects for 10 students and print them in the order of highest to lowest.

Solution:-

#include <stdio.h>

#include <string.h>

void main()

{

int rl,phy,che,ca,total;

float per;

char nm[20],div[10];

printf("Input the Roll Number of the student :");

scanf("%d",&rl);

printf("Input the Name of the Student :");

scanf("%s",nm);

printf("Input the marks of Physics, Chemistry and Computer Application : ");

scanf("%d%d%d",&phy,&che,&ca);

total = phy+che+ca;

per = total/3.0;

if (per>=60)

strcpy(div,"First");

else

if (per<60&&per>=48)

strcpy(div,"Second");

else

if (per<48&&per>=36)

strcpy(div,"Pass");

else

strcpy(div,"Fail");

printf("\nRoll No : %d\nName of Student : %s\n",rl,nm);

printf("Marks in Physics : %d\nMarks in Chemistry : %d\nMarks in Computer Application : %d\n",phy,che,ca);

printf("Total Marks = %d\nPercentage = %5.2f\nDivision = %s\n",total,per,div);

}

* Write a C program to demonstrate the use of Structure, to read the details of bank customer. The details are i) Cust\_ID, ii) Name iii) Age, iv) Address

v) Type of account.

Solution:-

#include <stdio.h>

struct customer

{

int account\_no;

char name[80];

int balance;

};

void accept(struct customer[], int);

void display(struct customer[], int);

int search(struct customer[], int, int);

void deposit(struct customer[], int, int, int);

void withdraw(struct customer[], int, int, int);

int main()

{

struct customer data[20];

int n, choice, account\_no, amount, index;

printf("Banking System\n\n");

printf("Number of customer records you want to enter? : ");

scanf("%d", &n);

accept(data, n);

do

{

printf("\nBanking System Menu :\n");

printf("Press 1 to display all records.\n");

printf("Press 2 to search a record.\n");

printf("Press 3 to deposit amount.\n");

printf("Press 4 to withdraw amount.\n");

printf("Press 0 to exit\n");

printf("\nEnter choice(0-4) : ");

scanf("%d", &choice);

switch (choice)

{

case 1:

display(data, n);

break;

case 2:

printf("Enter account number to search : ");

scanf("%d", &account\_no);

index = search(data, n, account\_no);

if (index == - 1)

{

printf("Record not found : ");

}

else

{

printf("A/c Number: %d\nName: %s\nBalance: %d\n",

data[index].account\_no, data[index].name,

data[index].balance);

}

break;

case 3:

printf("Enter account number : ");

scanf("%d", &account\_no);

printf("Enter amount to deposit : ");

scanf("%d", &amount);

deposit(data, n, account\_no, amount);

break;

case 4:

printf("Enter account number : ");

scanf("%d", &account\_no);

printf("Enter amount to withdraw : ");

scanf("%d", &amount);

withdraw(data, n, account\_no, amount);

}

}

while (choice != 0);

return 0;

}

void accept(struct customer list[80], int s)

{

int i;

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

{

printf("\nEnter data for Record #%d", i + 1);

printf("\nEnter account\_no : ");

scanf("%d", &list[i].account\_no);

fflush(stdin);

printf("Enter name : ");

gets(list[i].name);

list[i].balance = 0;

}

}

void display(struct customer list[80], int s)

{

int i;

printf("\n\nA/c No\tName\tBalance\n");

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

{

printf("%d\t%s\t%d\n", list[i].account\_no, list[i].name,

list[i].balance);

}

}

int search(struct customer list[80], int s, int number)

{

int i;

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

{

if (list[i].account\_no == number)

{

return i;

}

}

return - 1;

}

void deposit(struct customer list[], int s, int number, int amt)

{

int i = search(list, s, number);

if (i == - 1)

{

printf("Record not found");

}

else

{

list[i].balance += amt;

}

}

void withdraw(struct customer list[], int s, int number, int amt)

{

int i = search(list, s, number);

if (i == - 1)

{

printf("Record not found\n");

}

else if (list[i].balance < amt)

{

printf("Insufficient balance\n");

}

else

{

list[i].balance -= amt;

}

}

# Files:

* Write a C program to merge contents of two files into a third file. Print contents of the third file.

Solution:-

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*fp1 = fopen("file1.txt", "r");

FILE \*fp2 = fopen("file2.txt", "r");

FILE \*fp3 = fopen("file3.txt", "w");

char c;

if (fp1 == NULL || fp2 == NULL || fp3 == NULL)

{

puts("Could not open files");

exit(0);

}

while ((c = fgetc(fp1)) != EOF)

fputc(c, fp3);

while ((c = fgetc(fp2)) != EOF)

fputc(c, fp3);

printf("Merged file1.txt and file2.txt into file3.txt");

fclose(fp1);

fclose(fp2);

fclose(fp3);

return 0;

}

* Given a text file containing some text , write a program that prompts the user to input name of a text file , reads this file and outputs the number of vowels & number of words in the text.

Solution:-

#include <stdio.h>

int main() {

char line[150];

int vowels, consonant, digit, space;

vowels = consonant = digit = space = 0;

printf("Enter a line of string: ");

fgets(line, sizeof(line), stdin);

for (int i = 0; line[i] != '\0'; ++i) {

line[i] = tolower(line[i]);

if (line[i] == 'a' || line[i] == 'e' || line[i] == 'i' ||

line[i] == 'o' || line[i] == 'u') {

++vowels;

}

else if ((line[i] >= 'a' && line[i] <= 'z')) {

++consonant;

}

else if (line[i] >= '0' && line[i] <= '9') {

++digit;

}

else if (line[i] == ' ') {

++space;

}

} printf("Vowels: %d", vowels);

printf("\nConsonants: %d", consonant);

printf("\nDigits: %d", digit);

printf("\nWhite spaces: %d", space);

return 0;

}

* Write a C program to compare two text files. If corresponding characters are not same then print a message “Files are not same”. Else, display the contents of the either of the files in reverse order.

Solution:-

#include <stdio.h>

#include <stdlib.h>

int compareFile(FILE \* fPtr1, FILE \* fPtr2, int \* line, int \* col);

int main()

{

FILE \* fPtr1;

FILE \* fPtr2;

char path1[100];

char path2[100];

int diff;

int line, col;

printf("Enter path of first file: ");

scanf("%s", path1);

printf("Enter path of second file: ");

scanf("%s", path2);

fPtr1 = fopen(path1, "r");

fPtr2 = fopen(path2, "r");

if (fPtr1 == NULL || fPtr2 == NULL)

{

printf("\nUnable to open file.\n");

printf("Please check whether file exists and you have read privilege.\n");

exit(EXIT\_FAILURE);

}

diff = compareFile(fPtr1, fPtr2, &line, &col);

if (diff == 0)

{

printf("\nBoth files are equal.");

}

else

{

printf("\nFiles are not equal.\n");

printf("Line: %d, col: %d\n", line, col);

} fclose(fPtr1);

fclose(fPtr2);

return 0;

}

int compareFile(FILE \* fPtr1, FILE \* fPtr2, int \* line, int \* col)

{

char ch1, ch2;

\*line = 1;

\*col = 0;

do

{

ch1 = fgetc(fPtr1);

ch2 = fgetc(fPtr2);

if (ch1 == '\n')

{

\*line += 1;

\*col = 0;

}

if (ch1 != ch2)

return -1;

\*col += 1;

} while (ch1 != EOF && ch2 != EOF);

if (ch1 == EOF && ch2 == EOF)

return 0;

else

return -1;

}

* Write a program to sort the numbers sorted in a file.

Solution:-

#include <stdio.h>

int main(){

int temp, size;

int sort[] = {16,8,23,4,42,15};

size = sizeof(sort) / sizeof(int);

for(int j = 0; j < size; j++){

for(int i = 0; i < size; i++){

if(sort[i] > sort[i+1]){

temp = sort[i];

sort[i] = sort[i+1];

sort[i+1] = temp;

}

}

}

for(int p = 0; p < size; p++){

printf("%d ", sort[p]);

}

}