Module - 3.3

Name – Kishan Panara

# 1. Write a program to find out the max number from given array using function

# #include <stdio.h>

int find\_max(int arr[], int size) {

int max = arr[0];

for (int I = 1; I < size; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

int main()

{

int arr[] = {5, 9, 3, 2, 7, 1, 8, 4, 6};

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

int max\_num = find\_max(arr, size);

printf(“\n The maximum number in the array is %d”, max\_num);

return 0;

}

* **2. WAP of Addition, Subtraction, Multiplication and Division using Switch Case.(Must Be Menu Driven)**

#include <stdio.h>

int main() {

int choice, a, b, result;

printf(“\nEnter two numbers: “);

scanf(“%d %d”, &a, &b);

printf(“\nMENU:\n”);

printf(“\n 1.Addition\n2. Subtraction.\n3 Multiplication.\n4.Division”);

printf(“\n Enter your choice: “);

scanf(“%d”, &choice);

switch(choice)

{

case 1:

result = a + b;

printf(“\n%d + %d = %d\n”, a, b, result);

break;

case 2:

result = a – b;

printf(“\n%d - %d = %d\n”, a, b, result);

break;

case 3:

result = a \* b;

printf(“\n%d \* %d = %d\n”, a, b, result);

break;

case 4:

if(b == 0) {

printf(“\nCannot divide by zero”);

}

else

{

result = a / b;

printf(“\n%d / %d = %d\n”, a, b, result);

}

break;

default:

printf(“\nInvalid choice\n”);

break;

}

return 0;

}

* **3. WAP to find reverse of string using recursion**

#include <stdio.h>

#include <string.h>

void reverse(char \*);

int main()

{

char str[100];

printf(“\nEnter a string: “);

gets(str);

reverse(str);

printf(“\nReversed string is: %s”, str);

return 0;

}

void reverse(char \*s)

{

static int I = 0, j = 0;

if (\*s)

{

j++;

reverse(s + 1);

}

else

{

char rev[j + 1];

rev[j] = ‘\0’;

for (I = 0; I < j; i++)

{

rev[i] = \*(s – 1);

s--;

}

strcpy(s, rev);

}

}

* **4.WAP to find factorial using recursion**

#include <stdio.h>

int factorial(int n)

{

if(n == 1 || n == 0)

return 1;

else

return n \* factorial(n-1);

}

int main()

{

int num;

printf(“Enter a number: “);

scanf(“%d”, &num);

printf(“\nFactorial of %d is %d”, num, factorial(num));

return 0;

}

# 5. WAP to take two Array input from user and sort them in ascending or Descending order as per user’s choice

#include <stdio.h>

void ascendingSort(int arr[], int n)

{

int I, j, temp;

for (I = 0; I < n – 1; i++)

{

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

{

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

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

void descendingSort(int arr[], int n)

{

int I, j, temp;

for (I = 0; I < n – 1; i++)

{

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

{

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

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

int main() {

int arr1[100], arr2[100], n1, n2, I, choice;

printf(“\n Enter the number of elements in the first array: “);

scanf(“%d”, &n1);

printf(“\n Enter %d elements for the first array: “, n1);

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

scanf(“%d”, &arr1[i]);

}

printf(“\n Enter the number of elements in the second array: “);

scanf(“%d”, &n2);

printf(“\n Enter %d elements for the second array: “, n2);

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

{

scanf(“%d”, &arr2[i]);

}

printf(“\n Enter your choice for sorting order (1 for ascending, 2 for descending): “);

scanf(“%d”, &choice);

if (choice == 1) {

ascendingSort(arr1, n1);

ascendingSort(arr2, n2);

} else if (choice == 2) {

descendingSort(arr1, n1);

descendingSort(arr2, n2);

} else

{

printf(“\n Invalid choice!”);

return 0;

}

printf(“\n Sorted arrays:”);

printf(“First array: “);

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

printf(“%d “, arr1[i]);

}

printf(“\nSecond array: “);

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

printf(“%d “, arr2[i]);

}

return 0;

}

* **6. WAP to make addition, Subtraction and multiplication of two matrix using 2-D Array**

#include <stdio.h>

Int main()

{

Int I, j, k, rows1, cols1, rows2, cols2;

// Taking input for the dimensions of the first matrix

Printf(“Enter the number of rows and columns of the first matrix: “);

Scanf(“%d %d”, &rows1, &cols1);

// Taking input for the elements of the first matrix

Int mat1[rows1][cols1];

Printf(“Enter the elements of the first matrix:\n”);

For (I = 0; I < rows1; i++)

{

For (j = 0; j < cols1; j++)

{

Scanf(“%d”, &mat1[i][j]);

}

}

// Taking input for the dimensions of the second matrix

Printf(“\nEnter the number of rows and columns of the second matrix: “);

Scanf(“%d %d”, &rows2, &cols2);

Int mat2[rows2][cols2];

Printf(“\nEnter the elements of the second matrix:”);

For (I = 0; I < rows2; i++)

{

For (j = 0; j < cols2; j++)

{

Scanf(“%d”, &mat2[i][j]);

}

}

If (rows1 == rows2 && cols1 == cols2)

{

Int sum[rows1][cols1];

Printf(“Addition of the two matrices:\n”);

For (I = 0; I < rows1; i++)

{

For (j = 0; j < cols1; j++)

{

Sum[i][j] = mat1[i][j] + mat2[i][j];

Printf(“%d “, sum[i][j]);

}

Printf(“\n”);

}

} else

{

Printf(“\nAddition of two matrices not possible!”);

}

If (rows1 == rows2 && cols1 == cols2)

{

Int diff[rows1][cols1];

Printf(“\n Subtraction of the two matrices:”);

For (I = 0; I < rows1; i++)

{

For (j = 0; j < cols1; j++)

{

Diff[i][j] = mat1[i][j] – mat2[i][j];

Printf(“%d “, diff[i][j]);

}

printf(“\n”);

}

} else {

printf(“Subtraction of two matrices not possible!\n”);

}

if (cols1 == rows2) {

int prod[rows1][cols2];

printf(“Multiplication of the two matrices:\n”);

For (I = 0; I < rows1; i++) {

For (j = 0; j < cols2; j++) {

Prod[i][j] = 0;

For (k = 0; k < cols1; k++) {

Prod[i][j] += mat1[i][k] \* mat2[k][j];

}

Printf(“%d “, prod[i][j]);

}

Printf(“\n”);

}

} else {

printf(“Multiplication of two matrices not possible!\n”);

}

return 0;

}

* **7. WAP Find out length of string without using inbuilt function.**

#include <stdio.h>

int main() {

char str[100];

int length = 0;

printf(“\nEnter a string: “);

scanf(“%s”, str);

while (str[length] != ‘\0’)

{

length++;

}

printf(“\nLength of the string: %d”, length);

return 0;

}

* **8. WAP to reverse a string and check that the string is palindrome or not**

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

int I, len, flag = 0;

printf(“\n Enter a string: “);

fgets(str, 100, stdin);

len = strlen(str);

if (str[len-1] == ‘\n’)

{

str[len-1] = ‘\0’;

len--;

}

char rev\_str[100];

for (I = 0; I < len; i++)

{

rev\_str[i] = str[len-i-1];

}

rev\_str[len] = ‘\0’;

for (I = 0; I < len; i++)

{

if (str[i] != rev\_str[i])

{

flag = 1;

break;

}

}

if (flag)

{

printf(“\n%s is not a palindrome.”, str);

} else {

printf(“\n%s is a palindrome.”, str);

}

return 0;

}

* **9. Write a program of structure for five employee that provides the following**

**Information -print and display empno, empname, address and age**

#include <stdio.h>

struct Employee

{

int empno;

char empname[50];

char address[100];

int age;

};

Int main()

{

struct Employee emp[5];

int I;

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

printf(“\n Enter employee %d data:”, i+1);

printf(“Employee Number: “);

scanf(“%d”, &emp[i].empno);

printf(“Employee Name: “);

scanf(“%s”, emp[i].empname);

printf(“Employee Address: “);

scanf(“%s”, emp[i].address);

printf(“Employee Age: “);

scanf(“%d”, &emp[i].age);

}

printf(“\nEmployee Information:\n”);

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

printf(“\nEmployee %d:\n”, i+1);

printf(“Employee Number: %d\n”, emp[i].empno);

printf(“Employee Name: %s\n”, emp[i].empname);

printf(“Employee Address: %s\n”, emp[i].address);

printf(“Employee Age: %d\n”, emp[i].age);

}

return 0;

}