Module 3.2

Name – Kishan Panara

# 1. WAP to make simple calculator (operation include Addition, Subtraction, Multiplication, Division, modulo)

#include <stdio.h>

Int main()

{

char operator;

float num1, num2, result;

printf(“Enter operator (+, -, \*, /, %%) : “);

scanf(“%c”, &operator);

printf(“Enter two operands: “);

scanf(“%f %f”, &num1, &num2);

switch(operator)

{

case ‘+’:

result = num1 + num2;

printf(“%.2f + %.2f = %.2f”, num1, num2, result);

break;

case ‘-‘:

result = num1 – num2;

printf(“%.2f - %.2f = %.2f”, num1, num2, result);

break;

case ‘\*’:

result = num1 \* num2;

printf(“%.2f \* %.2f = %.2f”, num1, num2, result);

break;

case ‘/’:

result = num1 / num2;

printf(“%.2f / %.2f = %.2f”, num1, num2, result);

break;

case ‘%’:

result = (int)num1 % (int)num2;

printf(“%.2f %% %.2f = %.2f”, num1, num2, result);

break;

default:

printf(“Invalid operator”);

}

return 0;

}

* **2. WAP to swap two numbers without using third variable.**

#include <stdio.h>

int main()

{

int a, b;

printf(“Enter two numbers to swap: “);

scanf(“%d %d”, &a, &b);

// swapping

a = a + b;

b = a – b;

a = a – b;

printf(“\nAfter swapping, a = %d and b = %d”, a, b);

return 0;

}

* **3. WAP to find number is even or odd using ternary operator.**

#include <stdio.h>

int main()

{

int num;

printf(“\nEnter a number: “);

scanf(“%d”, &num);

(num % 2 == 0) ? printf(“%d is even.”, num) : printf(“%d is odd.”, num);

return 0;

}

* **4. WAP to show**

1. **Monday to Sunday using switch case**

#include <stdio.h>

int main()

{

int day;

printf(“\n Enter the day of the week (1-7): “);

scanf(“%d”, &day);

switch(day)

{

case 1:

printf(“\nMonday”);

break;

case 2:

printf(“\nTuesday”);

break;

case 3:

printf(“\nWednesday”);

break;

case 4:

printf(“\nThursday”);

break;

case 5:

printf(“\nFriday”);

break;

case 6:

printf(“\nSaturday”);

break;

case 7:

printf(“\nSunday”);

break;

default:

printf(“\nInvalid day”);

break;

}

return 0;

}

1. **Vowel or Consontant using switch case.**

#include <stdio.h>

int main()

{

char ch;

printf(“\n Enter a character: “);

scanf(“%c”, &ch);

switch(ch)

{

case ‘a’:

case ‘e’:

case ‘I’:

case ‘o’:

case ‘u’:

printf(“\n Vowel”);

break;

case ‘A’:

case ‘E’:

case ‘I’:

case ‘O’:

case ‘U’:

printf(“\n Vowel”);

break;

default:

printf(“Consonant”);

break;

}

return 0;

}

* **4. Looping programs:**

1. **WAP to print 972 to 897 using for loop.**

#include <stdio.h>

int main()

{

int I;

for (I = 972; I >= 897; i--) {

printf(“\n%d”, i);

}

return 0;

}

1. **WAP to take 10 no. Input from user and find out …**

**-How many Even numbers are there**

**-How many odd numbers are there**

**-Sum of even numbers**

**-Sum of odd numbers WAP to print table up to given numbers**

#include <stdio.h>

int main()

{

int num, i , evencount = 0, oddcount = 0, evensum = 0, oddsum = 0;

// Taking 10 numbers as input

printf(“Enter 10 numbers:\n”);

for (I = 1; I <= 10; i++)

{

scanf(“%d”, &num);

if (num % 2 == 0)

{

evencount++;

evensum += num;

}

else

{

oddcount++;

oddsum += num;

}

}

printf(“\nNumber of even numbers: %d”, evencount);

printf(“\nNumber of odd numbers: %d”, oddcount);

printf(“\nSum of even numbers: %d”, evensum);

printf(“\nSum of odd numbers: %d”, oddsum);

int n, j;

printf(“\nEnter a number to print its multiplication table: “);

scanf(“%d”, &n);

for (j = 1; j <= 10; j++)

{

printf(“\n%d x %d = %d", n, j, n\*j);

}

return 0;

}

* **5. WAP to print factorial of given number.**

#include <stdio.h>

int main() {

int num, i;

unsigned long long fact = 1;

printf("\nEnter a number : ");

scanf("%d", &num);

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

{

fact \*= i;

}

printf("\nFactorial of %d = %llu", num, fact);

return 0;

}

* **6. WAP to print number in reverse order e.g.: number = 64728 🡪 reverse = 82746**

#include <stdio.h>

int main() {

int num, reversed = 0;

printf(“Enter a number: “);

scanf(“%d”, &num);

while (num != 0)

{

reversed = reversed \* 10 + num % 10;

num /= 10;

}

printf(“\n Reversed number = %d”, reversed);

return 0;

}

* **7. Write a program to find out the max from given number (E.g., No: -1562. Max number is 6)**

#include <stdio.h>

int main() {

int num, max = 0;

printf(“\nEnter a number: “);

scanf(“%d”, &num);

while (num != 0)

{

int digit = num % 10;

if (digit > max)

{

max = digit;

}

num /= 10;

}

printf(“\n Maximum digit in the number = %d”, max);

return 0;

}

* **8. Write a program make a summation of given number (E.g., 1523 Ans: -11)**

#include <stdio.h>

int main() {

int num, sum = 0;

printf(“\nEnter a number: “);

scanf(“%d”, &num);

while (num != 0)

{

int digit = num % 10;

sum += digit;

num /= 10;

}

printf(“\nSummation of digits in the number = %d”, sum);

return 0;

}

* **9. Write a program you have to make a summation of first and last Digit. (E.g.,1234 Ans: - 5)**

#include <stdio.h>

int main() {

int num, first\_digit, last\_digit, sum;

printf(“\nEnter a number:”);

scanf(“%d”, &num);

while (first\_digit >= 10) {

first\_digit /= 10;

}

last\_digit = num % 10;

sum = first\_digit + last\_digit;

printf(“\nSum of first and last digits = %d”, sum);

return 0;

}

**Patterns :-**

1

1 0

1 0 1

1 0 1 0

1 0 1 0 1

#include <stdio.h>

int main()

{

int rows = 5;

for (int I = 1; I <= rows; i++)

{

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

{

if (j % 2 == 0)

{

printf(“0”);

}

else

{

printf(“1”);

}

}

printf(“\n”);

}

return 0;

}

**1**

**2 3**

**4 5 6**

**7 8 9 10**

**11 12 13 14 15**

#include <stdio.h>

int main()

{

int rows = 5;

int num = 1;

for (int I = 1; I <= rows; i++)

{

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

{

printf(“%d “, num);

num++;

}

printf(“\n”);

}

return 0;

}

**A**

**B C**

**D E F**

**G H I J**

**K L M N O**

#include <stdio.h>

int main()

{

int rows, I, j, count = 65;

printf(“Enter the number of rows: “);

scanf(“%d”, &rows);

for (I = 1; I <= rows; i++)

{

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

{

printf(“%c”, count);

count++;

}

printf(“\n”);

}

return 0;

}

**A**

**AB**

**ABC**

**ABCD**

**ABCDE**

#include <stdio.h>

int main()

{

int rows, I, j;

printf(“\nEnter the number of rows: “);

scanf(“%d”, &rows);

for (I = 1; I <= rows; i++)

{

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

{

printf(“%c”, ‘A’ + j – 1);

}

printf(“\n”);

}

return 0;

}

**\***

**\* \* \***

**\* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \***

#include <stdio.h>

int main() {

int rows = 5;

int I, j, k;

for (I = 1; I <= rows; i++) {

for (j = 1; j <= rows – I; j++) {

printf(“ “);

}

for (k = 1; k <= 2 \* I – 1; k++) {

printf(“\*”);

}

printf(“\n”);

}

return 0;

}

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \* \***

**\* \* \* \* \* \***

**\* \* \* \* \***

**\* \* \* \***

**\* \* \***

**\* \***

**\***

#include <stdio.h>

int main() {

int rows = 6;

int I, j;

for (I = 1; I <= rows; i++) {

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

printf(“\* “);

}

printf(“\n”);

if (I == rows) {

for (I = rows – 1; I >= 1; i--) {

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

printf(“\* “);

}

printf(“\n”);

}

}

}

return 0;

}

**Module – 3.3**

**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);

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]);

}

}

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;

}

* **10. Write a program of structure 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;

printf(“Enter employee number: “);

scanf(“%d”, &emp.empno);

printf(“Enter employee name: “);

scanf(“%s”, emp.empname);

printf(“Enter employee address: “);

scanf(“%s”, emp.address);

printf(“Enter employee age: “);

scanf(“%d”, &emp.age);

printf(“\nEmployee information:\n”);

printf(“Employee number: %d\n”, emp.empno);

printf(“Employee name: %s\n”, emp.empname);

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

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

return 0;

}

* **11.WAP to show difference between Structure and Union.**

#include <stdio.h>

#include <string.h>

struct Person {

char name[20];

int age;

float height;

};

union Data

{

int I;

float f;

char str[20];

};

int main()

{

struct Person person;

strcpy(person.name, “John”);

person.age = 30;

person.height = 1.75;

printf(“Person: %s, %d years old, %f meters tall\n”, person.name, person.age, person.height);

// Example of a union

union Data data;

data.i = 10;

printf(“Data: %d, %f, %s\n”, data.i, data.f, data.str);

data.f = 3.14;

printf(“Data: %d, %f, %s\n”, data.i, data.f, data.str);

strcpy(data.str, “Hello”);

printf(“Data: %d, %f, %s\n”, data.i, data.f, data.str);

return 0;

}