**FOCP\_ASSIGNMENT**

**//Assignment\_1**

//Q.1 wap to check for an Armstrong number

#include <stdio.h>

#include <math.h>

int main() {

int num, originalNum, remainder, n = 0, result = 0;

printf("Enter num: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

originalNum /= 10;

++n;

}

originalNum = num;

// Calculate the sum of digits raised to the power of n

while (originalNum != 0) {

remainder = originalNum % 10;

result += pow(remainder, n);

originalNum /= 10;

}

// Check if it's an Armstrong number

if (result == num)

printf("%d is an Armstrong number.\n", num);

else

printf("%d is not an Armstrong number.\n", num);

return 0;

}

//Q2. wap to calculate hcf to two integers

#include <stdio.h>

int main() {

int a,b,c;

printf("enter a and b : ");

scanf("%d %d", &a,&b);

int min=(a<b)?b:a;

for(int i=2;i<=min;i++){

if(a%i==0 && b%i==0){

c=i;

}

}

printf("%d", c);

return 0;

}

//Q3. WAP to subtract two integers without using Minus (-) operator. (Hint Bitwise operator)

#include <stdio.h>

int main()

{

int a,b;

printf("enter a and b :");

scanf("%d %d", &a ,&b);

int sub;

sub=~b;

sub=sub+1;

sub=sub+a;

printf("sub of %d and %d is %d", a,b,sub);

return 0;

}

/\*Q4. WAP to accept two integer numbers and swap them using 4 different methods in C language.\*/

#include <stdio.h>

int main()

{

//method 1;

int x,y;

printf("enter x and y :");

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

int z;

z=x+y;

x=z-x;

y=z-y;

printf("x is %d \ny is %d" ,x,y);

//method 2

int a,b,t;

printf("enter a and b :");

scanf("%d %d", &a , &b);

t=a;

a=b;

b=t;

printf("a is %d \nb is %d" ,a,b);

//method 3

int p,q,r;

printf("enter p and q :");

scanf("%d %d", &p , &q);

p=p-q;

q=p+q;

p=q-p;

printf("p is %d \nq is %d" ,p,q);

//method 4

int i,j;

printf("enter i and j :");

scanf("%d %d", &i , &j);

i=i\*j;

j=i/j;

i=i/j;

printf("i is %d \nj is %d" ,i,j);

return 0;

}

/\*Q5. WAP to check whether number is Perfect Number or not.

(To check perfect number, we have to find all divisors of that number and find their sum, if

sum of divisors is equal to number it means number is Perfect Number.)\*/

#include <stdio.h>

int main()

{

int x,y;

printf("enter a number :");

scanf("%d", &x);

int i,sum;

sum=0;

printf("factors are ");

for(i=1;i<x;i++){

 if(x%i==0){

    printf("%d\n", i);

    sum= sum+i;

 }

}

printf("sum is %d\n", sum);

    if(sum==x){

        printf("x is  perfect no\n");

    }

    if(sum!=x){

        printf("x is not a perfect no\n");

    }

return 0;

}

/\*Q6. WAP to accept a coordinate point in an XY coordinate system and determine in which

quadrant the coordinate point lies.

Test Data: 7 9

Expected Output: The coordinate point (7,9) lies in the First quadrant.\*/

#include <stdio.h>

int main()

{

int x,y;

printf("enter coordinate points (x,y) :");

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

if(x>0 && y>0){

    printf("the coordinate point (%d,%d) lies in the first quadrant", x, y);

}

if(x<0 && y>0){

    printf("the coordinate point (%d,%d) lies in the second quadrant", x, y);

}

if(x<0 && y<0){

    printf("the coordinate point (%d,%d) lies in the third quadrant", x, y);

}

if(x>0 && y<0){

    printf("the coordinate point (%d,%d) lies in the fourth quadrant", x, y);

}

if(x==0||y==0){

printf("this point is either on the x-axis or y-axis ");

}

return 0;

}

/\*Q7. WAP for Binary to Decimal conversion & Decimal to Binary for a given number as per user’s choice.\*/

#include<stdio.h>

#include<math.h>

int bintodec();

void dectobin();

int main(){

int choice;

printf("Choose conversion:\n1. Binary to Decimal\n2. Decimal to Binary\n");

scanf("%d", &choice);

switch (choice) {

case 1:

bintodec();

break;

case 2:

dectobin();

break;

default:

printf("Invalid choice.\n");

break;

}

return 0;

}

int bintodec(){

//binary to decimal conversion

long num;

int dec=0,rem=0,place=0;

printf("enter a binary no \n");

scanf("%ld", &num);

printf("\n binary equivalent of %ld in ", num);

while(num){

rem=num%10;

if(rem){

dec=dec+(pow(2,place));

}

num=num/10;

place++;

}

printf("decimal equivalent is %d\n",dec);

return dec;

}

void dectobin(){

int no,bin=0,rem=0,place=1;

printf("enter a decimal no \n");

scanf("%d", &no);

printf("\n decimal equivalent of %d in ", no);

while(no){

rem=no%2;

no=no/2;

bin=bin+(rem\*place);

place = place\*10;

}

printf("binary equivalent is %d\n",bin);

}

/\*Q8. WAP to print below mentioned pattern:

1

01

101

0101

10101\*/

#include <stdio.h>

int main()

{

int i,j,n;

printf("enter no of rows(n):  ");

scanf("%d", &n);

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

    for(j=0;j<=i;j++){

        if((i+j)%2==0){

            printf("1");

        }

        else{

            printf("0");

        }

    }

    printf("\n");

}

return 0;

}

/\*Q9. WAP to print following Pyramid:

0 0

01 01

010 010

0101 0101

0101001010\*/

#include <stdio.h>

void print\_pyramid(int rows) {

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

// Printing 0 and 1

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

printf("%d", j % 2);

}

//space printing

for (int j = 0; j < (rows - i - 1) \* 2; j++) {

printf(" ");

}

// Printing trailing part(0s and 1s)

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

printf("%d", j % 2);

}

printf("\n");

}

for (int i = 0; i < 2 \* rows; i++) {

printf("%d", i % 2);

}

printf("\n");

}

int main() {

int num\_rows = 5;

print\_pyramid(num\_rows);

return 0;

}

/\*Q10. WAP to print Pascal’s Triangle.Assignment 1- Basic C programming\*/

#include <stdio.h>

int main()

{

int i,j,space,value,n;

printf("enter no of rows(n):  ");

scanf("%d", &n);

value=1;

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

    for(space=1;space<=(n-i-1);space++){

       printf(" ");

    }

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

    printf("%2d",value);

    value=value\*(i-j)/(j+1);

}

printf("\n");

value = 1;

}

return 0;

}

**// Assignment \_2**

//Q1. WAP to increase every student mark by 5 & then print the updated array.

#include <stdio.h>

int main() {

int a[10];

printf("enter elements in array: ");

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

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

}

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

a[i]=a[i]+5;

}

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

printf("marks of student\_%d is %d\n", i,a[i]);

}

return 0;

}

//Q2. WAP to print grade of students as per their marks given in an array. (>=75-- A grade, 74 to 60--B Grade, 59 to 40--C grade below 40--D grade).

#include <stdio.h>

int main() {

int a[10],i;

printf("enter elements in array: ");

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

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

}

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

printf("grade of student\_%d with marks %d is ", (i+1),a[i]);

if(a[i]>=75){

printf("A\n");

}

else if(a[i]>=60 && a[i]<74){

printf("B\n");

}

else if(a[i]>=40 && a[i]<59){

printf("C\n");

}

else {

printf("D\n");

}

}

return 0;

}

//Q3. WAP to find who scored first “99” in an array marks.

#include <stdio.h>

int main() {

int a[4],i,found=0;

printf("enter elements in array: ");

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

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

}

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

if(a[i]==99){

found=1;

printf("student\_%d scored 99 first ", (i+1));

break;

}

}

if(!found){

printf("noone scored 99 marks ");

}

return 0;

}

//Q4. WAP to find Who & how many students have scored 99 in an array Marks.

#include <stdio.h>

int main() {

int a[5],i,found=0,student;

student=0;

printf("enter elements in array: ");

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

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

}

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

if(a[i]==99){

found=1;

printf("student\_%d scored 99 first\n ", (i+1));

student++;

}

}

if(!found){

printf("no one scored 99 marks ");

}

printf("total no of students scored 99 marks are : %d\n", student);

return 0;

}

//Q5. WAP to find sum of all scores in Marks array.

#include <stdio.h>

int main() {

int marks[5],i;

int sum=0;

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

printf("marks of %dth student is ", i+1);

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

}

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

sum=sum+marks[i];

}

printf("sum is %d", sum);

return 0;

}

//Q6. WAP to find average score of the Marks array.

#include <stdio.h>

int main() {

int marks[5],i;

int sum=0;

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

printf("marks of %dth student is ", i+1);

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

}

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

sum=sum+marks[i];

}

int average = sum/5;

printf("average is %d", average);

return 0;

}

//Q7. WAP to check whether score is even or odd in an array.

#include <stdio.h>

int main() {

int marks[5],i;

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

printf("marks of %dth student is ", i+1);

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

}

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

if(marks[i]%2==0){

printf("score of %dth student is even\n", i+1);

}

else {

printf("score of %dth student is odd\n", i+1);

}

}

return 0;

}

//Q8. WAP to find maximum & minimum score in the Marks array.

#include <stdio.h>

int main() {

int a[5],i,max,min;

printf("enter elements in array: ");

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

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

}

max=a[0];

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

if (max<a[i]){

max=a[i];

}

}

printf("max is %d\n", max);

min=a[0];

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

if (min>a[i]){

min=a[i];

}

}

printf("min is %d", min);

return 0;

}

//Q9. WAP to find a peak element which is not smaller than its neighbors.

#include <stdio.h>

int main() {

int a[5],i;

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

printf("marks of %dth student is ", i+1);

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

}

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

if(a[i-1]<a[i] && a[i+1]<a[i]){

printf("peak element is %d \n", a[i]);

}

}

return 0;

}

//Q10. WAP to count prime numbers in an array.

#include <stdio.h>

int main() {

int a[5], i, j, f, n;

int primeCount = 0;

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

printf("enter %dth element: ", i + 1);

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

}

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

n = a[i];

f = 0;

if (n <= 1) {

f = 1;

} else {

for (j = 2; j \* j <= n; j++) {

if (n % j == 0) {

f = 1;

break;

}

}

}

if (f == 0) {

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

primeCount++;

}

}

printf("Total prime numbers: %d\n", primeCount);

return 0;

}

/\*Q11. WAP to implement Insert -Front, any position in between & end in an array. Print the array before insert & after insert.\*/

#include <stdio.h>

int main() {

int A[6], i,posn,n,element;

printf("enter array size : ");

scanf("%d", &n);

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

printf("enter %dth element : ", i);

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

}

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

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

}

printf("enter element for insertion and posn: " );

scanf("%d %d",&element, &posn);

n++;

for(i=n;i>=posn;i--){

A[i]=A[i-1];

}

A[posn-1]=element;

printf("array after insertion is :\n");

for(i=0;i<6;i++){

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

}

return 0;

}

/\*Q12. WAP to implement delete-Front, any position in between & end in an array. Print the array before delete & after delete.\*/

#include <stdio.h>

int main() {

int A[5], i,posn;

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

printf("enter %dth element : ", i);

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

}

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

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

}

printf("enter posn for deletion : " );

scanf("%d", &posn);

for(i=posn-1;i<5;i++){

A[i]=A[i+1];

}

printf("array after deletion is :\n");

for(i=0;i<4;i++){

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

}

return 0;

}

/\*Q13. Given an array, the task is to cyclically rotate the array clockwise by one time. Examples: Input: arr[] = {1, 2, 3, 4, 5} Output: arr[] = {5, 1, 2, 3, 4} Input: arr[] = {2, 3, 4, 5, 1} Output: {1, 2, 3, 4, 5} \*/

#include <stdio.h>

int main() {

int arr[6], i,j,n,element;

printf("enter array size : ");

scanf("%d", &n);

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

printf("enter %dth element : ", i);

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

}

i=n-1;

j=n-2;

while(i>0){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

i--;

j--;

}

printf("Reversed array:\n");

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

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

}

return 0;

}

/\*Q14. Given an array of n integers. The task is to print the duplicates in the given array.

If there are no duplicates then print -1.

Examples:

Input: {2, 10,10, 100, 2, 10, 11,2,11,2}

Output: 2 10 11

Input: {5, 40, 1, 40, 100000, 1, 5, 1}

Output: 5 40 1 \*/

#include <stdio.h>

int main() {

    int i,j,a[6],flag=1;

    printf("Enter 6 elements of array\n");

    for (i=0;i<6;i++)

    {scanf("%d", &a[i]); }

    printf("Array: ");

    for (i=0;i<6;i++)

    {printf("%d ", a[i]); }

    printf("\n");

    for (i=0;i<6;i++)

    {for (j=i+1;j<6;j++)

    {if (a[i] == a[j])

    {printf("%d is a duplicate\n", a[i]);

    flag=0;

    break;}}}

    if(flag==1)

    {printf("-1"); }

return 0;

}

**// Assignment\_3**

#include <stdio.h>

#include <math.h>

int main(){

int a,b,c;

int sum,sub,prod,div;

printf("enter a and b respectively : ");

scanf("%d %d", &a,&b);

sum = a+b;

sub = a-b;

prod = a\*b;

div = a/b;

int operator;

printf("enter\n 1 for sum\n 2 for sub\n 3 for prod\n 4 for div\n 5 for sqrt\n 6 for log\n 7 for natural log\n ");

printf("enter operator : ");

scanf("%d", &operator);

if(operator == 1){

printf("%d\n", sum);

}

else if(operator == 2){

printf("%d\n", sub);

}

else if(operator == 3){

printf("%d\n", prod);

}

else if(operator == 4){

printf("%f\n", div);

}

else if(operator == 5 && a>0){

printf("%f\n", sqrt(a));

}

else if(operator == 6 && a>0){

printf("%f\n", log10(a));

}

else if(operator == 7 && a>0){

printf("%f\n", log(a));

}

else {

printf("invalid case");

}

return 0;

}

**// Assignment\_4**

#include <stdio.h>

int main(){

   char ch\_1,ch\_2;

   printf("press:\n r for rock\n p for paper\n s for scissors\n");

   printf("user\_1 move : ");

   scanf(" %c", &ch\_1);

   printf("user\_2 move : ");

   scanf(" %c", &ch\_2);

   if(ch\_1 == ch\_2){

       printf("tie! play again");

   }

   else if(ch\_1 == 'r' && ch\_2 == 'p'){

       printf("paper wins!\n");

       printf("user\_2 won\n");

   }

   else if(ch\_1 == 'p' && ch\_2 == 'r'){

       printf("paper wins!\n");

       printf("user\_1 won\n");

   }

   else if(ch\_1 == 'p' && ch\_2 == 's'){

       printf("scissors wins!\n");

       printf("user\_2 won\n");

   }

   else if(ch\_1 == 's' && ch\_2 == 'p'){

       printf("scissors wins1\n");

       printf("user\_1 won\n");

   }

   else if(ch\_1 == 'r' && ch\_2 == 's'){

       printf("rock wins!\n");

       printf("user\_1 won\n");

   }

   else if(ch\_1 == 's' && ch\_2 == 'r'){

       printf("rock wins!\n");

       printf("user\_2 won\n");

   }

   else {

       printf("default case\n");

   }

    return 0;

}

**// Assignment\_6**

/\* Assignment 5

The Hangman Game is one of the most famous games played

on computers.

The Rules of the game are as follows:

1. There is given a word with omitted characters and you

need to guess the characters to win the game.

2. Only 3 chances are available and if you win the Man

survives or Man gets hanged.

So, it is the game can be easily designed in C language with

the basic knowledge of if-else statements, loops, and some

other basic statements. The code of the game is easy, short,

and user-friendly.\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#define MAX\_TRIES 6

const char \*words[] = {"python", "hangman", "programming", "challenge", "openai"};

void print\_hangman(int tries) {

printf("\n");

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

printf(" | |\n");

if (tries < 6) printf(" | O\n");

if (tries < 5) printf(" | /|\\\n");

if (tries < 4) printf(" | / \\\n");

printf(" |\n");

}

int main() {

srand(time(NULL));

const char \*word = words[rand() % (sizeof(words) / sizeof(words[0]))];

int word\_length = strlen(word);

char guessed[word\_length + 1];

int tries = MAX\_TRIES;

int correct\_guesses = 0;

for (int i = 0; i < word\_length; i++) {

guessed[i] = '\_';

}

guessed[word\_length] = '\0';

printf("Welcome to Hangman!\n");

while (tries > 0 && correct\_guesses < word\_length) {

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

print\_hangman(tries);

printf("Enter a letter: ");

char input;

scanf(" %c", &input);

int found = 0;

for (int i = 0; i < word\_length; i++) {

if (word[i] == input && guessed[i] == '\_') {

guessed[i] = input;

found = 1;

correct\_guesses++;

}

}

if (!found) {

tries--;

printf("Incorrect! You have %d tries left.\n", tries);

} else {

printf("Good guess!\n");

}

}

if (correct\_guesses == word\_length) {

printf("Congratulations! You've guessed the word: %s\n", word);

} else {

printf("Sorry, you've run out of tries. The word was: %s\n", word);

}

return 0;

}

**// Assignment\_6**

/\* Tic Tac Toe Game

The working of tic tac toe game is same as traditional tic tac toe

having following components:

\* Objective: To be the first to make a straight line with

either ‘X’ or ‘O’.

\* Game Board: The board consists of a 3×3 matrix-like

structure, having 9 small boxes.

\* The computer: Since it is a two-player game each player

gets one chance alternatively. i.e.; first player1 than

player2.

\* Moves: The computer starts the game with O. After that

player makes moves alternatively.

\* Winning: You win by making your symbol in a row or

diagonal or column. Also, as a part of strategy you need to

block your opponent from forming a straight line while

making of your own. \*/

#include <stdio.h>

#include <string.h>

void printboard();

int checkwin();

char board[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

int main() {

int player = 1, input, status;

status = -1;

printboard();

while (status == -1) {

player = (player % 2 == 0) ? 2 : 1;

char mark = (player == 1) ? 'X' : 'O';

printf("Player %d's turn: ", player);

scanf("%d", &input);

if (input < 1 || input > 9 || board[input] == 'X' || board[input] == 'O') {

printf("Invalid input. Please try again.\n");

continue;

}

board[input] = mark;

printboard();

status = checkwin();

if (status == 1) {

printf("Player %d wins!\n", player);

} else if (status == 0) {

printf("It's a draw!\n");

}

player++;

}

return 0;

}

void printboard() {

printf("\*-- Tic Tac Toe --\*\n");

printf("%c | %c | %c\n", board[1], board[2], board[3]);

printf("\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\n");

printf("%c | %c | %c\n", board[4], board[5], board[6]);

printf("\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\n");

printf("%c | %c | %c\n", board[7], board[8], board[9]);

printf(" | | \n");

}

int checkwin() {

if (board[1] == board[2] && board[2] == board[3]) return 1;

if (board[4] == board[5] && board[5] == board[6]) return 1;

if (board[7] == board[8] && board[8] == board[9]) return 1;

if (board[1] == board[4] && board[4] == board[7]) return 1;

if (board[2] == board[5] && board[5] == board[8]) return 1;

if (board[3] == board[6] && board[6] == board[9]) return 1;

if (board[1] == board[5] && board[5] == board[9]) return 1;

if (board[3] == board[5] && board[5] == board[7]) return 1;

// Check for a draw

int count = 0;

for (int i = 1; i <= 9; i++) {

if (board[i] == 'X' || board[i] == 'O') {

count++;

}

}

if (count == 9) {

return 0; // Draw

}

return -1;

}