ASSIGNMENT 1

Q1.WAP to check whether a given number is Armstrong or not.

#include <stdio.h>

int main(){

int num,original\_num,remainder,flag=0;

printf("Enter a three-digit number integer=");

scanf("%d",&num);

original\_num=num;

while(original\_num!=0){

remainder=original\_num%10;

flag+=remainder\*remainder\*remainder;

original\_num/=10;

}

if (flag == num)

printf("%d is an armstrong number",num);

else

printf("%d is not a armstrong number",num);

return 0;

}

Q2.WAP to read two integers and print their HCF.

#include <stdio.h>

int main()

{

int num1,num2,i,hcf;

printf("enter number 1=");

scanf("%d",&num1);

printf("enter number 2=");

scanf("%d",&num2);

for(i=1;i<=num1&&i<=num2;i++){

if(num1%i==0 && num2%i==0){

hcf=i;

}

}

printf("HCF of %d and %d is %d",num1,num2,hcf);

return 0;

}

Q3. WAP to subtract two integer without using minus(-) operator.

#include <stdio.h>

int main()

{

int a,b;

a=5;

b=3;

printf("Bitwise NOT=~%d = %d\n",a,~a);

return 0;

}

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

SWAPPING #1

#include <stdio.h>

int main() {

int a,b;

printf("Enter two integers: ");

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

a = a + b;

b = a - b;

a = a - b;

printf("After swapping: a = %d, b = %d\n",a,b);

printf("Hence ,number is:%d %d\n",a,b);

return 0;

}

SWAPPING #2

#include <stdio.h>

int main() {

int a,b;

printf("Enter two integers (non-zero): ");

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

a = a \* b;

b = a / b;

a = a / b;

printf("After swapping: a= %d, b= %d\n",a,b);

return 0;

}

SWAPPING #3

#include <stdio.h>

int main() {

int a,b;

printf("Enter two integers (non-zero): ");

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

a = a ^ b;

b = a ^ b;

a = a ^ b;

printf("After swapping: a= %d, b= %d\n",a,b);

return 0;

}

SWAPPING #4

#include <stdio.h>

int main() {

int a,b;

printf("Enter two integers (non-zero): ");

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

int temp = a;

a =b;

b =temp;

printf("After swapping: a= %d, b= %d\n",a,b);

return 0;

}

Q6. WAP to accept a coordinate point in a XY coordinate system and determine in which quadrant the coordinate point lies.

#include <stdio.h>

void determineQuadrant(int x, int y) {

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

printf("The point (%d, %d) lies in Quadrant I.\n", x, y);

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

printf("The point (%d, %d) lies in Quadrant II.\n", x, y);

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

printf("The point (%d, %d) lies in Quadrant III.\n", x, y);

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

printf("The point (%d, %d) lies in Quadrant IV.\n", x, y);

} else if (x == 0 && y != 0) {

printf("The point (%d, %d) lies on the Y-axis.\n", x, y);

} else if (x != 0 && y == 0) {

printf("The point (%d, %d) lies on the X-axis.\n", x, y);

} else {

printf("The point (%d, %d) is at the Origin.\n", x, y);

}

}

int main() {

int x, y;

printf("Enter the X coordinate: ");

scanf("%d", &x);

printf("Enter the Y coordinate: ");

scanf("%d", &y);

determineQuadrant(x, y);

return 0;

}

Q7.WAP for binary to decimal conversion and decimal to binary for a given number as per user’s choice.

#include <stdio.h>

#include <math.h>

int binaryToDecimal(int binary) {

int decimal = 0, base = 1, remainder;

while (binary > 0) {

remainder = binary % 10;

decimal = decimal + remainder \* base;

binary = binary / 10;

base = base \* 2;

}

return decimal;

}

int decimalToBinary(int decimal) {

int binary = 0, base = 1;

while (decimal > 0) {

int remainder = decimal % 2;

binary = binary + remainder \* base;

decimal = decimal / 2;

base = base \* 10;

}

return binary;

}

int main() {

int choice, number;

printf("Choose conversion type:\n");

printf("1. Binary to Decimal\n");

printf("2. Decimal to Binary\n");

printf("Enter your choice (1 or 2): ");

scanf("%d", &choice);

if (choice == 1) {

printf("Enter a binary number: ");

scanf("%d", &number);

printf("Decimal equivalent: %d\n", binaryToDecimal(number));

} else if (choice == 2) {

printf("Enter a decimal number: ");

scanf("%d", &number);

printf("Binary equivalent: %d\n", decimalToBinary(number));

} else {

printf("Invalid choice!\n");

}

return 0;

}

Q8. WAP to print below mentioned pattern;

1

01

101

0101

10101

#include <stdio.h>

int main() {

int rows = 5;

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

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

if ((i + j) % 2 == 0) {

printf("1");

} else {

printf("0");

}

}

printf("\n");

}

return 0;

}

Q9. Wap to print following pyramid:

00

01 01

010 010

0101 0101

01010 01010

#include <stdio.h>

int main() {

int rows = 5;

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

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

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

}

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

printf(" ");

}

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

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

}

printf("\n");

}

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

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

}

printf("\n");

return 0;

}

Q10. WAP to print pascal’s triangle.

#include <stdio.h>

int main() {

int rows;

printf("Enter the number of rows for Pascal's Triangle: ");

scanf("%d", &rows);

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

int n=1;

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

printf(" ");

}

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

printf("%d ", n);

n = n\* (i - j) / (j + 1);

}

printf("\n");

}

return 0;

}