ASSIGNMENT -1

//QUES.1.

//WAP to check whether a given is Armstrong or not…

#include <stdio.h>

#include <math.h>

int main() {

int num, originalNum, remainder, n = 0;

float result = 0.0;

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

originalNum /= 10;

++n;

}

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += pow(remainder, n);

originalNum /= 10;

}

if ((int)result == num) {

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

} else {

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

}

return 0;

}

//Ques.2.

//WAP to read two integers and print their HCF (Highest Common Factor)…

#include <stdio.h>

int findHCF(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int main() {

int num1, num2;

printf("Enter two integers: ");

scanf("%d %d", &num1, &num2);

int hcf = findHCF(num1, num2);

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

return 0;

}

//QUES.3.

//WAP to subtract two integers without using Minus (-) operator…

#include <stdio.h>

int subtract(int a, int b) {

while (b != 0) {

int c = (~a) & b;

a = a ^ b;

b = c<< 1;

}

return a;

}

int main() {

int num1, num2;

printf("Enter two integers: ");

scanf("%d %d", &num1, &num2);

int result = subtract(num1, num2);

printf("Result of %d - %d is %d\n", num1, num2, result);

return 0;

}

//QUES.4.

/\*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;

}

/\*QUES.5.

WAP to check whether number is Perfect Number or not…

\*/

#include <stdio.h>

int Perfect(int num) {

int sum = 0;

for (int i = 1; i <= num / 2; i++) {

if (num % i == 0) {

sum += i;

}

}

return sum == num;

}

int main() {

int number;

printf("Enter an integer: ");

scanf("%d", &number);

if (Perfect(number) && number > 0) {

printf("%d is a Perfect Number.\n", number);

} else {

printf("%d is not a Perfect Number.\n", number);

}

return 0;

}

/\*QUES.6.

WAP to accept a coordinate point in an 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;

}

/\*QUES.7.

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 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;

}

/\*QUES.8.

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;

}

/\*QUES.9.  
WAP to print following Pyramid:  
0 0  
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;

}

/\*QUES.10.

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;

}