**Assignment- 1**

**Name – Ishita Yadav**

**Roll number – 24csu084**

Ques1- WAP to check whether a given is Armstrong or not

Sol:

#include <stdio.h>

//Programme to check whether a number is armstrong or not

int main() {

int num,remainder,sum=0,num1;

printf("Enter a number :");

scanf("%d",&num);

num1=num;

while (num1!=0) {

remainder=num1%10;

sum=sum+(remainder\*remainder\*remainder);

num1=num1/10;

}

if (sum==num)

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

else

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

return 0;

}

Ques 2: WAP to read two integers and print their HCF (Highest Common Factor).

Sol:

#include <stdio.h>

//Programme to print HCF of two integers

int main() {

int num1,num2,i,HCF;

printf("Enter first number :");

scanf("%d",&num1);

printf("Enter second number :");

scanf("%d",&num2);

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

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

HCF=i;

}

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

return 0;

}

Ques 3: WAP to subtract two integers without using Minus (-) operator.

Sol:

#include <stdio.h>

int subtract(int a, int b) {

while (b != 0) {

// Find the carry and shift it left

int carry = (~a) & b;

// Perform subtraction using XOR

a = a ^ b;

// Update b to the carry

b = carry << 1;

}

return a;

}

int main() {

int num1, num2, result;

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

result = subtract(num1, num2);

printf("Subtraction of %d and %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

Sol:

#include <stdio.h>

//Programme to swap two numbers using four different method's

int main() {

int num1,num2;

printf("Enter first number :");

scanf("%d",&num1);

printf("Enter second number :");

scanf("%d",&num2);

printf("%d , %d after swapping values become %d , %d\n",num1,num2,num2,num1);

int num3;

num3=num2;

num2=num1;

num1=num3;

printf("Values after swapping %d , %d\n",num3,num2);

}

Ques 5: 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

Sol:

#include <stdio.h>

//Programme to check whether number is perfect or not.

int main() {

int num,i,sum=0,sum1;

printf("Enter a number :");

scanf("%d",&num);

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

while(num%i==0) {

printf("%d, ",i);

sum=sum+i;

i=i+1;

}

}

sum1=sum-num;

if (sum1==num)

printf("\n%d is a perfect number",num);

else

printf("\n%d is not a perfect number",num);

return 0;

}

---------------------------------------------------------------------------------

Ques 6: 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.

Sol:

#include <stdio.h>

//Programme to find quadrant's

int main() {

int x,y;

printf("Enter X-cordinate :");

scanf("%d",&x);

printf("Enter Y-cordinate :");

scanf("%d",&y);

if (x>0 && y>0)

printf("The cordinate point (%d,%d) lies in First-quadrant",x,y);

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

printf("The cordinate point (%d,%d) lies in Second-quadrant",x,y);

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

printf("The cordinate point (%d,%d) lies in Third-quadrant",x,y);

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

printf("The cordinate point (%d,%d) lies in Fourth-quadrant",x,y);

else

printf("The cordinate point (%d,%d) is at origin",x,y);

return 0;

}

----------------------------------------------------------------------------------

Ques 7: WAP for Binary to Decimal conversion & Decimal to Binary for a given number as per user’s choice

Sol:

#include <stdio.h>

#include <math.h>

long long binaryToDecimal(long long n);

long long decimalToBinary(int n);

int main() {

int choice;

long long binary, decimal;

int number;

printf("Choose an option:\n");

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

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

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

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter a binary number: ");

scanf("%lld", &binary);

decimal = binaryToDecimal(binary);

printf("%lld in binary = %lld in decimal\n", binary, decimal);

break;

case 2:

printf("Enter a decimal number: ");

scanf("%d", &number);

binary = decimalToBinary(number);

printf("%d in decimal = %lld in binary\n", number, binary);

break;

default:

printf("Invalid choice!\n");

}

return 0;

}

long long binaryToDecimal(long long n) {

long long decimal = 0;

int i = 0, remainder;

while (n != 0) {

remainder = n % 10;

n /= 10;

decimal += remainder \* pow(2, i);

++i;

}

return decimal;

}

long long decimalToBinary(int n) {

long long binary = 0;

int remainder, i = 1;

while (n != 0) {

remainder = n % 2;

n /= 2;

binary += remainder \* i;

i \*= 10;

}

return binary;

}

----------------------------------------------------------------------------------

Ques 8: Q8. WAP to print below mentioned pattern:  
1  
01  
101  
0101  
10101

Sol:

#include <stdio.h>

int main() {

int i;

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

switch (i) {

case 1:

printf("1\n");

break;

case 2:

printf("0 1\n");

break;

case 3:

printf("1 0 1\n");

break;

case 4:

printf("0 1 0 1\n");

break;

case 5:

printf("1 0 1 0 1");

break;

}

}

return 0;

}

----------------------------------------------------------------------------------

Ques 9: WAP to print following Pyramid:  
0 0  
01 01  
010 010  
0101 0101  
0101001010

Sol: #include <stdio.h>

void printPyramid(int n) {

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

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

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

}

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

printf(" ");

}

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

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

}

printf("\n");

}

}

int main() {

int n = 5;

printPyramid(n);

return 0;

}

----------------------------------------------------------------------------------

Ques10: WAP to print Pascal’s Triangle.

Sol:

#include <stdio.h>

long factorial(int n) {

long result = 1;

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

result \*= i;

}

return result;

}

int main() {

int rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

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

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

printf(" ");

}

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

printf("%ld ", factorial(i) / (factorial(j) \* factorial(i - j)));

}

printf("\n");

}

return 0;

}