ASSIGNMENT 1

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

**CODE-**

#include <stdio.h>

#include <math.h>

int main()

{

int num,x,rem,n=0,sum=0;

printf("Enter a number-\n");

scanf("%d",&num);

x=num;

while(x!=0)

{x/=10;

++n; }

x=num;

while(x!=0)

{rem=x%10;

sum+=pow(rem,n);

x/=10; }

if((int)sum==num)

{ printf("ARMSTRONG NUMBER"); }

else

{printf("NOT AN ARMSTRONG NUMBER"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

Q2. WAP to read two integers and print their HCF (Highest Common Factor).

**CODE-**

#include <stdio.h>

int main()

{ int x,y,i,min;

printf("Enter two numbers\n");

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

min=(x>y)?y:x;

for(i=min;i>0;i--)

{ if(x%i==0 && y%i==0)

{printf("The hcf is-%d",i);

break; } }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

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

**CODE-**

#include <stdio.h>

int main()

{

int a,b,c;

printf("Enter a and b-\n");

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

c=~b;

c=c+1;

c=c+a;

printf("The difference of %d and %d is %d", a,b,c);

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

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

**CODE-**

#include <stdio.h>

int main() {

int a, b,n;

printf("Enter two integers: \n");

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

printf("Select the method\nMethod 1(Three variables)\nMethod 2(Without three variable)\nMethod 3(bitwise XOR)\nMethod 4(pointers)\n");

scanf("%d",&n);

printf("Before Swapping: a = %d, b = %d\n", a, b);

switch(n)

{case 1:{//method 1

int c;

c=a;

a=b;

b=c;

printf("After Swapping (Method 1): a = %d, b = %d\n", a, b);

break;}

case 2:{//method 2

a=a+b;

b=a-b;

a=a-b;

printf("After Swapping (Method 2): a = %d, b = %d\n", a, b);

break;}

case 3:{//method 3

a = a ^ b;

b = a ^ b;

a = a ^ b;

printf("After Swapping (Method 3): a = %d, b = %d\n", a, b);

break;}

case 4:{//method 4

int \*p = &a;

int \*q = &b;

\*p = \*p + \*q;

\*q = \*p - \*q;

\*p = \*p - \*q;

printf("After Swapping (Method 4): a = %d, b = %d\n", a, b);

break;}

default: printf("Invalid choice"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

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.)

**CODE-**

#include <stdio.h>

int main() {

int n, i,sum = 0;

printf("Enter a number: \n");

scanf("%d", &n);

for (i=1;i<=(n/2);i++)

{ if (n%i==0)

{sum+=i; }}

if (sum==n)

{ printf("%d is a perfect number.\n",n);}

else

{ printf("%d is not a perfect number.\n",n); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

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.

**CODE-**

#include <stdio.h>

int main() {

int x, y;

printf("Enter the x,y coordinates\n");

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

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

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

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

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

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

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

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

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

} else {

printf("The coordinate point (0,0) lies at origin");

}

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

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

**CODE-**

#include <stdio.h>

#include <math.h>

int main() {

int choice, n;

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

printf("Enter your choice: \n");

scanf("%d", &choice);

if (choice == 1)

{ int binary, decimal = 0, base = 1, rem, flag;

printf("Enter a binary number: ");

scanf("%d", &binary);

n = binary;

while (n> 0) {

int rem = n % 10;

if (rem != 0 && rem != 1) {

flag = 0;

break;

}

n= n/ 10;

}

if(flag==0)

{printf("Enter valid binary number\n"); ;

}

while (binary > 0 && flag!=0) {

rem = binary % 10;

decimal = decimal + rem \* base;

binary = binary / 10;

base = base \* 2;

}

printf("Binary %d in decimal is: %d\n", n, decimal);

}

else if (choice == 2)

{ int decimal, binary = 0, base = 1, rem;

printf("Enter a decimal number: ");

scanf("%d", &decimal);

n = decimal;

while (decimal > 0) {

rem= decimal % 2;

binary = binary + rem \* base;

decimal = decimal / 2;

base = base \* 10;

}

printf("Decimal %d in binary is: %d\n", n, binary);

}

else

{printf("Invalid choice. Please enter 1 or 2.\n"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

**CODE-**

#include <stdio.h>

int main()

{ int i,j;

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

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

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

{printf("1"); }

else

{printf("0"); } }

printf("\n"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

Q9. WAP to print following Pyramid:  
0 0  
01 01  
010 010  
0101 0101  
0101001010

**CODE-**

#include <stdio.h>

int main()

{ int i,j,k,x;

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

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

{ if(j%2==0)

{printf("0"); }

else

{printf("1"); } }

x=2\*(4-i);

for(j=0;j<x;j++)

{printf(" "); }

for(k=0;k<=i;k++)

{ if(k%2==0)

{printf("0"); }

else

{printf("1"); } }

printf("\n"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated

Q10. WAP to print Pascal’s Triangle.

**CODE-**

#include <stdio.h>

int main()

{ int i,j,k,n;

printf("Enter the no. of rows\n");

scanf("%d",&n);

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

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

{printf(" "); }

int z=1;

for(k=0;k<=i;k++)

{printf("%d ",z);

z=z\*(i-k) / (k+1); }

printf("\n"); }

return 0;

}

**OUTPUT-**

A screenshot of a computer

Description automatically generated