**Assignment 1 (Basic C Programming)**

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

#include <stdio.h>

#include <math.h>

int main() {

int num = 153, sum = 0, temp, digits = 0, rem;

temp = num;

while (temp > 0) {

digits++;

temp /= 10;

}

temp = num;

while (temp > 0) {

rem = temp % 10;

sum += pow(rem, digits);

temp /= 10;

}

if (sum == num)

printf("Armstrong\n");

else

printf("Not Armstrong\n");

return 0;

}

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

Factor).

#include <stdio.h>

int main() {

int a = 56, b = 98, temp;

while (b != 0) {

temp = b;

b = a % b;

a = temp;

}

printf("HCF: %d\n", a);

return 0;

}

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

**Bitwise operator)**

#include <stdio.h>

int main() {

int a = 15, b = 7;

while (b != 0) {

int borrow = (~a) & b;

a = a ^ b;

b = borrow << 1;

}

printf("Result: %d\n", a);

return 0;

}

**Q4. WAP to accept two integer numbers and swap them using 4**

**different methods in C language**.

#include <stdio.h>

int main() {

//Using a third variable

int c = 10, d = 20, temp;

temp = c;

c = d;

d = temp;

printf("c = %d, d = %d\n", c, d);

//Without using a third variable (Arithmetic addition and

subtraction)

int e = 10, f = 20;

e = e + f;

f = e - f;

e = e - f;

printf("\ne = %d, f = %d\n", e, f);

//Without using a third variable (Arithmetic multiplication and

division)

int g = 10, h = 20;

g = g \* h;

h = g / h;

g = g / h;

printf("g = %d, h = %d\n", g, h);

//Using bitwise XOR operator

int a = 10, b = 20;

a = a ^ b;

b = a ^ b;

a = a ^ b;

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

return 0;

}

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

#include <stdio.h>

int main() {

int num = 28, sum = 0, i;

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

if (num % i == 0) sum += i;

}

if (sum == num)

printf("Perfect Number\n");

else

printf("Not a Perfect Number\n");

return 0;

}

**Q6. WAP to accept a coordinate point in an XY coordinate system and**

**determine in which quadrant the coordinate point lies.**

#include <stdio.h>

int main() {

int x = 7, y = 9;

if (x > 0 && y > 0)

printf("The coordinate point (%d,%d) lies in the First

quadrant.\n", x, y);

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

printf("The coordinate point (%d,%d) lies in the Second

quadrant.\n", x, y);

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

printf("The coordinate point (%d,%d) lies in the Third

quadrant.\n", x, y);

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

printf("The coordinate point (%d,%d) lies in the Fourth

quadrant.\n", x, y);

else

printf("The coordinate point (%d,%d) lies on the origin or an

axis.\n", x, y);

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 binaryToDecimal(int binary) {

int decimal = 0, base = 1;

while (binary > 0) {

int lastDigit = binary % 10;

decimal += lastDigit \* base;

base \*= 2;

binary /= 10;

}

return decimal;

}

void decimalToBinary(int decimal) {

int binary[32], i = 0;

while (decimal > 0) {

binary[i++] = decimal % 2;

decimal /= 2;

}

for (int j = i - 1; j >= 0; j--) {

printf("%d", binary[j]);

}

printf("\n");

}

int main() {

int choice, num;

printf("Enter 1 for Binary to Decimal, 2 for Decimal to Binary: ");

scanf("%d", &choice);

printf("Enter the number: ");

scanf("%d", &num);

if (choice == 1)

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

else if (choice == 2) {

printf("Binary: ");

decimalToBinary(num);

}

return 0;

}

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

**1**

**01**

**101**

**0101**

**10101**

#include <stdio.h>

int main() {

int n = 5;

for (int i = 1; i <= n; 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:**

**0 0**

**01 01**

**010 010**

**0101 0101**

**0101001010**

#include <stdio.h>

int main() {

int n = 5;

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

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

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

}

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

printf(" ");

}

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

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

}

printf("\n");

}

return 0;

}

**Q10. WAP to print Pascal’s Triangle.**

#include <stdio.h>

int main() {

int n = 5, coef = 1;

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

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

printf(" ");

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

if (j == 0 || i == 0)

coef = 1;

else

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

printf("%4d", coef);

}

printf("\n");

}

return 0;

}