

PSCP Assignment-2

By: Adarsh Rao

Section C

832068

Q1) A Dudeney number is a positive integer that equals the cube of its sum of digits. For example, $512 = 8 * 8 * 8$ is a Dudeney number because $8 = 5 + 1 + 2$. Similarly, $4913 = 17 * 17 * 17$ is Dudeney as $17 = 4 + 9 + 1 + 3$. Write a program to print the Dudeney numbers below 10000.

```
#include <iostream>

int sumofdigits(int num);

int main(){
for(int i{10}; i < 10000; i++){

if (i == sumofdigits(i)*sumofdigits(i)*sumofdigits(i)){

std::cout << i << '\n';
}
}

}

int sumofdigits(int num){

int sum{};
while (num != 0) {
sum = sum + num%10;
num = num/10;

}
return sum;
}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.1
512
4913
5832
```

Q2) A positive integer is called square-free if it is not divisible by the square of any prime number. For Example, $98 = 2 \times 49$, $99 = 9 \times 11$, $100 = 4 \times 25$ are not square-free, whereas $42 = 2 \times 3 \times 7$ and $102 = 2 \times 3 \times 17$ are square-free. Write a program to find all square-free numbers below 1000.

```
#include <iostream>
#include <cmath>
```

```
int main(){
```

```
int yes{};
```

```
for(int i=1;i<=1000;++i){
```

```
yes=0;
```

```
for(int j=2;j<=sqrt(i);++j){
```

```
if(i % (j*j)==0){
```

```
yes=1;
```

```
break;
```

```
}
```

```
}
```

```
if(yes==0){
```

```
std::cout<<i<< '\n';
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.2
```

```
1
2
3
5
6
7
10
11
13
14
15
17
19
21
22
23
26
29
30
```

Q3) The Fibonacci series 0,1,1,2,3,5,8,13... begins with the terms 0 and 1 and has the property that each succeeding term is the sum of the two preceding terms. Write a program to print all numbers that do not appear in the Fibonacci series upto the given 'n'.

```
#include <iostream>
```

```
int main(){  
int n{};  
int x{};  
int fib2{};  
int a{};  
int b{1};  
int fib{0};
```

```
std::cout<<"Enter the number till which Non-fibonacci numbers are to be printed: ";  
std::cin>>n;
```

```
while(fib<=n){
```

```
fib=a+b;  
a=b;  
b=fib;  
fib2=a+b;
```

```
for(x=fib+1;x<fib2;x++){  
if(x<=n){  
std::cout<<x<<"\n";  
}  
else{  
break;  
}  
}  
}  
return 0;  
}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.3  
Enter the number till which Non-fibonacci numbers are to be printed: 10
```

```
4  
6  
7  
9  
10
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ █
```

Q4) An organization wants to transmit data securely over telephone, but they are concerned that their phone may be tapped. All the data are transmitted as five digit numbers. The organization wants to hide the actual five digit number by encoding so that it can be transmitted more securely. The encoding method is as follows. Each digit in the number will be replaced by sum of digits upto that digit modulus 10. Write a program which takes (as input) a series of five digit numbers and print the encoded numbers. For example, if the number is 23849, the encoded number is computed as follows $2\%10=2$, $(2+3)\%10=5$, $(2+3+8)\%10=3$, $(2+3+8+4)\%10=7$, $(2+3+8+4+9)\%10=6$, Hence the encoded number is 25376.

```
#include <iostream>
#include <cmath>

int sumofdigits(int num);
int noofdigits(int num);

int main(){

    int x{};
    std::cout << "Enter a 5 digit number: ";
    std::cin >> x;

    if (noofdigits(x) == 5){

        int sum{};

        for(int i{}; i < 5; i++){

            sum = sum + sumofdigits(x)%10 * pow(10,i);
            x = x/10;

        }

        std::cout << "The encoded value is: " << sum << "\n";
    }

    else{

        std::cout << "Please enter a 5 Digit number!";

    }

}

int sumofdigits(int num){
```

```
int sum{};

while ( num != 0){
sum = sum + num%10;
num = num/10;
}

return sum;

}
```

```
int noofdigits(int num){

int count{};

while ( num != 0){
count++;
num = num/10;
}

return count;

}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.4
Enter a 5 digit number: 23849
The encoded value is: 25376
```

—

Q5) There are 9000 people in a town whose population increases by 15% each year. Write a program that displays the annual population and determines the number of years it will take for the population to surpass 50000.

```
#include <iostream>
#include <iomanip>

int main(){

    int i{};
    float population = 9000.00;

    do {
        population = population + (0.1500)*population;
        i++;
    }
    while(population + 0.15*population < 50000);

    std::cout << std::fixed;
    // You can use this to set the precision. Changing the precision to something higher will end up in
    // decimal values, so I decided to set the precision to 0 since it's population..
    std::cout << std::setprecision(0);
    std::cout << "Final Population: " << population << "\n";
    std::cout << "Number of Years taken: " << i << "\n";

}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.5
Final Population: 48152
Number of Years taken: 12
```

Q6) Write a program to accept a number and find sum of its individual digits repeatedly till the result is a single digit. For example, if the given number is 4687, sum of digits = $4+6+8+7=25=2+5=7$. Output should be 7.

```
#include <iostream>

int sumofdigits(int num);

int main(){
    int num{};
    std::cout << "Enter a number: ";
    std::cin >> num;
    while (num/10 != 0){
        num = sumofdigits(num);
    }

    int sumofdigits(int num){

        int sum{};
        while ( num != 0){
            sum = sum + num%10;
            num = num/10;
        }
        std::cout << sum << '\n';

        return sum;

    }
}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.6
Enter a number: 4687
25
7
```


Q7) Write a menu driven program which has following options

- 1-Factorial of a number
- 2-Prime or not
- 3-Odd or even
- 4. Nth Fibonacci number
- 5-Exit

Once a menu item is selected the appropriate option should be taken and once this option is finished, the menu should reappear .Unless the user selects the Exit option the program should continue work.

```
#include <iostream>
```

```
int factorial();  
int primeornot();  
int oddeven();  
int fibonacci();
```

```
int main(){
```

```
int i{};
```

```
while (i == 0){
```

```
int y{};
```

```
std::cout << "Enter what you want to do.\n1. Factorial of a number.\n2. Prime or not.\n3. Odd or  
Even.\n4. Nth Fibonacci number.\n5. Exit \n";
```

```
std::cin >> y;
```

```
switch (y){
```

```
case 1:
```

```
factorial();
```

```
break;
```

```
case 2:
```

```
primeornot();
```

```
break;
```

```
case 3:
```

```
oddeven();
```

```
break;
```

```
case 4:
```

```
fibonacci();
```

```

break;
case 5:
i = 1;
break;
default:
std::cout << "Please enter a valid option! \n";
}

}

}

int factorial(){

int x{};
std::cout << "Enter a number: ";
std::cin >> x;

int product{1};

for(int i{1}; i <= x; i++){

product=product*i;

}

std::cout << "The factorial is: " << product << '\n';
return 0;
}

int primeornot(){

bool isPrime = true;
int n{};
std::cout << "\nEnter a positive integer: ";
std::cin >> n;

if (n == 0 || n == 1) {
isPrime = false;
}
else {
for (int i{}; i <= n / 2; ++i) {
if (n % i == 0) {
isPrime = false;
break;
}
}
}
if (isPrime){

```

```
std::cout << n << " is a prime number\n";
}
else{
std::cout << n << " is not a prime number\n";
}

return 0;
}
```

```
int oddeven(){

int x{};
std::cout << "Enter a number: ";
std::cin >> x;

if (x%2==0){

std::cout << x << " is Even\n";

}

else if (x%2 !=0){

std::cout << x << " is Odd\n";

}
return 0;
}
```

```
int fibonacci() {
int x{};
int num1{};
int num2{1};
int nextTerm{0};

std::cout << "Enter the number of terms: ";
std::cin >> x;

std::cout << "The series is: ";

for (int i = 1; i <= x; ++i) {
if(i == 1) {
std::cout << num1 << ", ";
continue;
}
if(i == 2) {
std::cout << num2 << ", ";
continue;
}
```

```

}
nextTerm = num1 + num2;
num1 = num2;
num2 = nextTerm;
std::cout << nextTerm << ", ";
}
std::cout << '\n';
return 0;
}

```

adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2\$./PSCP_Assignment_2.7

Enter what you want to do.

1. Factorial of a number.
2. Prime or not.
3. Odd or Even.
4. Nth Fibonacci number.
5. Exit

36

Please enter a valid option!

Enter what you want to do.

1. Factorial of a number.
2. Prime or not.
3. Odd or Even.
4. Nth Fibonacci number.
5. Exit

1

Enter a number: 5

The factorial is: 120

Enter what you want to do.

1. Factorial of a number.
2. Prime or not.
3. Odd or Even.
4. Nth Fibonacci number.
5. Exit

3

Enter a number: 7

7 is Odd

Enter what you want to do.

1. Factorial of a number.
2. Prime or not.
3. Odd or Even.
4. Nth Fibonacci number.
5. Exit

5

adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2\$ █

Q8) Write a program that accepts a year written as a four-digit numeral and outputs the year written in Roman numerals. Important Roman numerals are V –5 , X-10 , L-50 , C-100, D-500 and M-1000

```
#include <iostream>
```

```
int main(){  
int year{};  
int rem{};  
std::cout<<"Enter the year (4 Digits or less): ";  
std::cin>>year;
```

```
if (year < 10000){
```

```
if(year>=1000){  
rem = year/1000;  
if (rem < 5){  
for(int i{1}; i <= rem; i++){  
std::cout<<"M";  
}  
}
```

```
if (rem >= 5 && rem < 9){
```

```
std::cout << "V̄";  
int d = rem - 5;  
for(int a=1;a<=d;a++){  
std::cout<<"M";  
}  
}
```

```
if (rem == 9){
```

```
std::cout << "IX̄";
```

```
}
```

```
year-=rem*1000;  
}
```

```
if(year>=100){  
rem=year/100;  
if(rem < 4){  
for(int j=1; j<=rem; j++){  
std::cout<<"C";  
}  
}
```

```
if(rem==4){  
std::cout<<"CD";
```

```

}
if((rem>=5)&&(rem<9)){
std::cout<<"D";
int d = (year/100)-5;
for(int a=1;a<=d;a++){
std::cout<<"C";
}
}
if(rem==9){
std::cout<<"CM";
}
year-=rem*100;
}
if(year>=10){
rem=year/10;
if(rem<4){
for(int k=1;k<=rem;k++){
std::cout<<"X";
}
}
if(rem==4)
std::cout<<"XL";
if((rem>=5)&&(rem<9)){
std::cout<<"L";
int n=(year/10)-5;
for(int m=1;m<=n;m++){
std::cout<<"X";
}
}
if(rem==9){
std::cout<<"XC";
}
}

```

```

year-=rem*10;
}

```

```

if(year>0){
rem=year;
if(rem<4){
for(int a=1;a<=rem;a++){
std::cout<<"I";
}
}
if(rem == 4)
std::cout<<"IV";
if((rem >= 5) && (rem < 9)){
std::cout<<"V";
}
}

```

```

int m= year-5;
for(int n=1; n<=m; n++){
std::cout<<"I";
}
}
if(rem==9){
std::cout<<"IX";
}
}

std::cout << '\n';
}

else{

std::cout << "Please enter a valid year!" << '\n';

}
return 0;
}

```

```

adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.8
Enter the year (4 Digits or less): 7777
V̄MMDCCLXXVII
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.8
Enter the year (4 Digits or less): 2021
MMXXI
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.8
Enter the year (4 Digits or less): 1947
MCMXLVII
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.8
Enter the year (4 Digits or less): 736
DCCXXXVI
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.8
Enter the year (4 Digits or less): 88888
Please enter a valid year!adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$

```

Q9) Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example $153 = (1*1*1) + (5*5*5) + (3*3*3)$.

```
#include <iostream>
#include <cmath>

int findsum(int num);

int main(){

    int endno{500};
    for (int i{1}; i<=endno; i++){

        if (i == findsum(i)){

            std::cout << i << "\n";
        }
    }
}

int findsum(int num){

    int sum{};
    while (num != 0){

        sum = sum + std::pow(num%10,3);
        num = num/10;

    }
    return sum;
}
```

```
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.9
1
153
370
371
407
adrao@adrao-g14:~/Documents/C/PSCP Assignment/PSCP Assignment 2$
```


Q10) Write a program to print the following output:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
 1 3 3 1
  1 2 1
   1 1
    1
```

```
    *
   * *
  * * *
 * * * *
* * * * *
 * * * *
  * * *
   * *
    *
```

```
/*
```

So I have a factorial function. I'm going to use this to make $nCr = n! / ((n-r)! * r!)$

```
*/
```

```
#include <iostream>
```

```
int factorial(int num);
```

```
int printSpace(int space);
```

```
int main(){
```

```
    int rows{4};
```

```
    int i{0};
```

```
    while(i < rows){
```

```
        int space = (rows) - i ;
```

```
        printSpace(space);
```

```
        for (int k{}; k <= i ; k++){
```

```
            std::cout << factorial(i)/(factorial(k)*factorial(i-k)) << " ";
        }
```

```
        std::cout << "\n";
```

```
        i++;
```

```
    }
```

```
    while (i >= 0){
```

```
        int space = rows - i;
```

```
        printSpace(space);
```

```
        for (int k{}; k <= i ; k++){
```

```
std::cout << factorial(i)/(factorial(k)*factorial(i-k)) << " ";  
}
```

```
std::cout << "\n";  
i--;
```

```
}
```

```
while(i < rows){  
int space = (rows) - i ;  
printSpace(space);
```

```
for (int k{}; k <= i ; k++){
```

```
std::cout << "* ";  
}
```

```
std::cout << "\n";
```

```
i++;  
}
```

```
while (i >= 0){
```

```
int space = rows - i;  
printSpace(space);  
for (int k{}; k <= i ; k++){
```

```
std::cout << "* ";  
}
```

```
std::cout << "\n";  
i--;
```

```
}
```

```
}
```

```
int printSpace(int space){
```

```
for (int i{}; i<=space; i++){
```

```
std::cout << " ";
```

```
}  
return 0;  
  
}
```

```
int factorial(int num){  
int product{1};  
for(int y{};y < num; y++){  
  
product = product*(num-y);  
  
}  
return product;  
  
}
```

```
adrao@adrao-g14: ~/Documents/C/PSCP Assignment/PSCP Assignment 2$ ./PSCP_Assignment_2.10
```

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
1 3 3 1  
1 2 1  
1 1  
1
```

```
 *  
 * *  
 * * *  
 * * * *  
 * * * * *  
 * * * * *  
 * * * *  
 * * *  
 * *  
 *
```