**Program 1:** Find Factors of a Positive Integer:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

std::cout << "Factors = ";

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

if (pNum % i == 0) {

std::cout << i << " "<<endl;

}

}

return 0;

}

**Output Snip:**

**Program 2:** Reverse a Positive Integer:

#include <iostream>

int main() {

int pNum,digit, reverseNum = 0;;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

while (pNum > 0) {

digit= pNum % 10;

reverseNum = reverseNum \* 10 + digit;

pNum /= 10;

}

std::cout << "Reverse = " << reverseNum<<endl;

return 0;

}

**Output Snip:**

**Program 3:** Sum of Numbers between Two Positive Integers:

#include <iostream>

int main() {

int pN, pM;

std::cout << "Enter two positive integers (N and M): ";

std::cin >> pN >> pM;

int sum = 0;

for (int i = pN; i <= pM; ++i) {

sum += i;

}

std::cout << "Sum = " << sum<<endl;

return 0;

}

**Output Snip:**

**Program 4:** Prime Numbers between Two Positive Integers:

#include <iostream>

int main() {

int pN, pM;

std::cout << "Enter two positive integers (N and M): ";

std::cin >> pN >> pM;

std::cout << "Prime Numbers = "<<endl;

for (int num = pN; num <= pM; ++num) {

bool isPrime = true;

if (num <= 1) {

isPrime = false;

} else {

for (int i = 2; i \* i <= num; ++i) {

if (num % i == 0) {

isPrime = false;

break;

}

}

}

if (isPrime) {

std::cout << num << " "<<endl;

}

}

return 0;

}

**Output Snip:**

**Program 5:** Find the First Digit in a Positive Integer:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

int firstDigit;

while (pNum!=0) {

firstDigit =pNum%10;

pNum=pNum/10;

}

std::cout << "First Digit = " <<firstDigit<<endl;

return 0;

}

**Output Snip:**

**Program 6:** Sum of Digits of a Positive Integer:

#include <iostream>

int main() {

int pNum, sum = 0,digit;;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

while (pNum > 0) {

digit=pNum%10;

sum=sum+digit;

pNum=pNum/10;

}

std::cout << "Sum of digits = " << sum<<endl;

return 0;

}

**Output Snip:**

**Program 7:** Sum of Digits of an Integer (Positive or Negative):

#include <iostream>

int main() {

int Number;

std::cout << "Enter an integer: ";

std::cin >> pNum;

int sum = 0;

if(pNum<0)

pNum=-1\*pNum;

while (pNum!=0) {

sum += Number % 10;

Number /= 10;

}

std::cout << "Sum of digits = " << sum<<endl;

return 0;

}

**Output Snip:**

**Program 8:** Product of Digits of a Positive Integer:

#include <iostream>

int main() {

int Number;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

int product = 1;

while (pNum!=0) {

product= product\*(pNum % 10);

pNum /= 10;

}

std::cout << "Product of digits = " << product<<endl;

return 0;

}

**Output Snip:**

**Program 9:** Check if a Positive Integer is a Palindrome:

#include <iostream>

int main() {

int pNum,pNumCopy, reverse = 0;

std::cout << "Enter a positive integer: ";

std::cin >>pNum;

pNumCopy= pNum;

while (pNumCopy> 0) {

reverse = reverse \* 10 + pNumCopy % 10;

pNumCopy /= 10;

}

if (pNum== reverse) {

std::cout << "A palindrome."<<endl;

} else {

std::cout << "Not a palindrome."<<endl;

}

return 0;

}

**Output Snip:**

**Program 10:** Check if a Positive Integer is a Perfect Number:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

int sum = 0;

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

if (pNum % i == 0) {

sum += i;

}

}

if (sum == pNum) {

std::cout << "A perfect number."<<endl;

} else {

std::cout << "Not a perfect number."<<endl;

}

return 0;

}

**Output Snip:**

Program 11: Check if a Positive Integer is an Armstrong Number:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

int num = pNum;

int numDigits = log10(pNum) + 1;

int sum = 0;

while (num > 0) {

int digit = num % 10;

sum += pow(digit, numDigits);

num /= 10;

}

if (sum == pNum) {

std::cout << "An Armstrong number."<<endl;

} else {

std::cout << "Not an Armstrong number.<endl;

}

return 0;

}

**Output Snip:**

**Program 12:** Find the Count of a Digit in a Given Positive Integer:

#include <iostream>

int main() {

int Number, Digit;

std::cout << "Enter a positive integer: ";

std::cin >> Number;

std::cout << "Enter a digit to count: ";

std::cin >> Digit;

int count = 0;

while (Number > 0) {

if (Number % 10 == Digit) {

count++;

}

Number /= 10;

}

std::cout << "Count of digit " << Digit << " in the number = " << count<<endl;

return 0;

}

**Output Snip:**

**Program 13:** Print Digits in a Positive Integer as Words:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

std::

string words[] = {"Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

std::cout << "Result = ";

int digit;

while (pNum > 0) {

digit = pNum % 10;

std::cout << words[digit] << " ";

pNum /= 10;

}

return 0;

}

**Output Snip:**

**Program 14:** Print a Positive Integer in Words:

#include <iostream>

int main() {

int pNum;

std::cout << "Enter a positive integer: ";

std::cin >> pNum;

std::string words[] = {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

std::string teens[] = {"", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

std::string tens[] = {"", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

if (pNum == 0) {

std::cout << "Zero";

} else {

if (pNum >= 1000) {

std::cout << words[pNum / 1000] << " Thousand ";

pNum %= 1000;

}

if (pNum >= 100) {

std::cout << words[pNum / 100] << " Hundred ";

pNum %= 100;

if (pNum > 0) {

std::cout << "and ";

}

}

if (pNum >= 20) {

std::cout << tens[pNum / 10] << " ";

pNum %= 10;

}

if (pNum > 10 && pNum < 20) {

std::cout << teens[pNum - 11] << " ";

pNum = 0;

}

if (pNum > 0) {

std::cout << words[pNum] << " ";

}

}

std::cout<<endl;

return 0;

}

**Output Snip:**