Rajshahi University of Engineering & Technology

CSE 2102: Sessional Based on CSE 2101

Lab Report 06

Dated: 24.04.18

Submitted to

Rizoan Toufiq

Assistant Professor

Dept. of Computer Science & Engineering

&

Instructor, CSE 2102

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

Experiment No. 3

Name of the Experiment: Algorithms, Number Theory and Cryptography.

1. EXPERIMENT [07]

Given an integer n, use the greedy algorithm to find the change for n cents using quarters, dimes, nickels, and pennies.

SOLUTION:

```
#include <iostream>

using namespace std;

int main () {
    int coin[4] = {25, 10, 5, 1};
    int coin_count[4] = {0};
    int n;
    cin >> n;

    for(int i = 0; i < 4; i++) {
        while (n >= coin[i]) {
            coin_count[i]++;
            n -= coin[i];
        }
    }
    cout << endl;
    cout    << coin_count[0] << " quarter\n"
            << coin_count[1] << " dime\n"
            << coin_count[2] << " nickel\n"
            << coin_count[3] << " penny\n";
}
```

OUTPUT:

```
Enter Cents to change: 65

2 quarter
1 dime
1 nickel
0 penny
```

```
Enter Cents to change: 47

1 quarter
2 dime
0 nickel
2 penny
```

2. EXPERIMENT [14]

Given a positive integer, determine whether it is prime.

SOLUTION:

```
#include <iostream>

using namespace std;

int main() {
  long long p, i;
  cout << "Input the number to check if it is prime or not: ";
  cin >> p;
  int flag = 1;
  for(i = 2; i * i <= p; i++) {
     if(p % i == 0) {
          flag = 0;
          break;
     }
  }
  if(1 == flag) {
      cout << "\nPrime." << endl;
  } else cout << "\nNot Prime, divisible by " << i << endl;
}
```

OUTPUT:

```
Input the number to check if it is prime or not: 179424673

Prime.
```

```
Input the number to check if it is prime or not: 47055833459

Prime.
```

```
Input the number to check if it is prime or not: 47055833458

Not Prime, divisible by 2
```

Discussion: Here, 179424673 and 47055833459 is the 10 millionth and 2 billionth prime number respectively. The code is effective enough to check any number in this range and even beyond.

3. EXPERIMENT [15]

Given a positive integer, determine whether it is Mersenne prime.

SOLUTION:

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

using namespace std;

int main() {
    unsigned long long p, i;
    cout << "Input a number to check if it creates a Mersenne "
         << "prime or not: ";
    cin >> p;
    int flag = 1;
    for(i = 2; i * i <= p; i++) {
        if(p % i == 0) {
            flag = 0;
            break;
        }
    }
    if(1 == flag) {
        cout << "P = " << p << " is Prime." << endl;

        long long n = pow(2, p) - 1;
        flag = 1;
        for(i = 2; i * i <= n; i++) {
            if(n % i == 0) {
                flag = 0;
                break;
            }
        }
        if(1 == flag) {
            cout << "N = " << n
                    << " is Mersenne Prime." << endl;
        } else cout << "N = " << n
                    << " is not Mersenne Prime, divisible by "
                    << i << endl;
    } else cout << "P = " << p << " is not Prime, divisible by "
                << i
                << "\nAnd no need to check for Mersenne prime."
                << endl;
}
```

OUTPUT:

```
Input a number to check if it creates a Mersenne prime or not: 61
P = 61 is Prime.
N = 2305843009213693951 is Mersenne Prime.
```

Discussion: Here, atmost Mersenne prime is determined for 61 as the power of 2 ($2^{61}-1$), though it gives a 19 digit prime. The next Mersenne prime has 27 digit and it is 2^{89}-1, is beyond this calculations – gives garbage value.

4. EXPERIMENT [16]

The polynomial f(n) = n^2-n+41 has the interesting property that f(n) is prime for all positive integers n not exceeding 40. Given a positive integer n, find the value of f(n) whether f(n) is prime or not.

SOLUTION:

```
#include <iostream>

using namespace std;

int func(int n) {
    return n * n - n + 41;
}

int main() {
    int p;
    cout << "Enter the value of n (n^2 - n + 41): ";
    cin >> p;

    p = func(p);

    int flag = 1;
    for(int i = 2; i * i <= p; i++) {
        if(p % i == 0) {
            flag = 0;
            break;
        }
    }
    if(1 == flag) {
        cout << p << " is a Prime number." << endl;
    } else cout << p << " is Not a Prime number." << endl;
}
```

OUTPUT:

```
Enter the value of n (n^2 - n + 41): 40
1601 is a Prime number.
```

```
Enter the value of n (n^2 - n + 41): 41
1681 is not a Prime number.
```

5. EXPERIMENT [17]

[Goldbach's Conjecture] Given an even integer n, find two prime number whether the sum of them is equal to n.

SOLUTION:

```
#include <iostream>

using namespace std;

bool isPrime(long long p) {
    int flag = 1;
    for(long long i = 2; i * i <= p; i++) {
        if(p % i == 0) {
            flag = 0;
            break;
        }
    }
    if(1 == flag) {
        return true;
    } else return false;
}
int main() {
    int flag = 0;
    long long p, i;
    cout << "Enter an Even Integer: ";
    cin >> p;
    cout << endl;
    for(i = 2; 2 * i < p; i++) {
        if(isPrime(i) && isPrime(p - i)) {
            cout << i << " + " << p - i << endl;
            flag = 1;
        }
    }
    if (0 == flag) cout << "Not Found." << endl;
}
```

OUTPUT:

```
Enter an Even Integer: 100

3 + 97
11 + 89
17 + 83
29 + 71
41 + 59
47 + 53
```

Discussion: More than one solution of couple-prime can be found for one given even integer.

6. EXPERIMENT [18]

Given an integer n, whether f(n) = n2+1 is prime or not.

SOLUTION:

```
#include <iostream>

using namespace std;

int func(int n) {
    return n * n + 1;
}

int main() {
    int p;
    cout << "Enter the value of n (n^2 + 1): ";
    cin >> p;

    p = func(p);

    int flag = 1, i;
    for(i = 2; i * i <= p; i++) {
        if(p % i == 0) {
            flag = 0;
            break;
        }
    }
    if(1 == flag) {
        cout << p << " is Prime." << endl;
    } else cout << p << " is Not Prime, divisible by " << i <<
"." << endl;
}
```

OUTPUT:

```
Enter the value of n (n^2 + 1): 30
901 is Not Prime, divisible by 17.
```

```
Enter the value of n (n^2 + 1): 40
1601 is Prime.
```

6. EXPERIMENT [19]

[The Twin Prime Conjecture] Given a positive number n, whether it is prime or not. If n is prime, check whether n and n+2 are Twin primes or not.

SOLUTION:

```
#include <iostream>

using namespace std;

bool isPrime(long long p) {
    int flag = 1;
    for(int i = 2; i * i <= p; i++) {
        if(p % i == 0) {
            flag = 0;
            break;
        }
    }
    if(1 == flag) {
        return true;
    } else return false;
}

int main() {
    long long n;
    cout << "Enter the value of n: ";
    cin >> n;
    cout << endl;
    if(isPrime(n)) {
        cout << n << " is Prime." << endl;
        if(isPrime(n + 2)) {
            cout << n + 2 << " is Prime Too." << endl;
            cout << n << " and " << n + 2 << " are Twin prime." << endl;
        } else cout << n + 2 << " is not Prime." << endl;
    } else {
         cout << n << " is not Prime" << endl;
         cout << "No need to check for n + 2 is prime or not." << endl;
    }
}
```

OUTPUT:

```
Enter the value of n: 101
101 is Prime.
103 is Prime Too.
101 and 103 are Twin prime.
```

```
Enter the value of n: 13
13 is Prime.
15 is not Prime.
They are not Twin Prime.
```

```
Enter the value of n: 15
15 is not Prime
No need to check for n + 2 is prime or not.
```