

Computer Programming

Quiz1

Apr. 11, 2024



Problem 1 (pr1.c)

- Write a program to take a lowercase alphabet from the user and determine whether the input alphabet is a consonant or a vowel.
 - ✓ Implement using a *switch* statement.
 - ✓ The input will be a lowercase alphabet.
 - ✓ Consonants and vowels are defined as follows:
 - Consonants: ‘b’, ‘c’, ‘d’, ‘f’, ‘g’, ‘h’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘p’, ‘q’, ‘r’, ‘s’, ‘t’, ‘v’, ‘w’, ‘x’, ‘y’, ‘z’
 - Vowels: ‘a’, ‘e’, ‘i’, ‘o’, ‘u’
 - ✓ If the input is not a lowercase alphabet, print “Invalid input”.

Problem 1 (pr1.c)

- **Program output**

```
[ohyong@cse Quiz1_s12]$ vi pr1.c
[ohyong@cse Quiz1_s12]$ gcc pr1.c -o pr1
[ohyong@cse Quiz1_s12]$ ./pr1
Enter a lowercase alphabet: e
Vowel
```

```
[ohyong@cse Quiz1_s12]$ ./pr1
Enter a lowercase alphabet: k
Consonant
```

```
[ohyong@cse Quiz1_s12]$ ./pr1
Enter a lowercase alphabet: @
Invalid input
```

Problem 2 (pr2.c)

- Write a program to determine whether a given year entered by the user is a leap year or not.
 - ✓ Implement using an *if* statement.
 - ✓ If the entered year is a leap year, print “Leap year”, otherwise print “Not a leap year (Normal year)”.
 - ✓ A leap year is defined as follows:
 - A year that is divisible by 4 is called a leap year.
 - Among them, a year that is divisible by 100 is a normal year.
 - However, a year that is divisible by 400 from a normal year is called a leap year.
 - For example, the year 2024 is a leap year because it is evenly divisible by 4.

Problem 2 (pr2.c)

- **Program output**

```
[ohyong@cse Quiz1_s12]$ vi pr2.c
[ohyong@cse Quiz1_s12]$ gcc pr2.c -o pr2
[ohyong@cse Quiz1_s12]$ ./pr2
Enter a year: 2024
Leap year

[ohyong@cse Quiz1_s12]$ ./pr2
Enter a year: 2025
Not a leap year(Normal year)

[ohyong@cse Quiz1_s12]$ ./pr2
Enter a year: 2100
Not a leap year(Normal year)
```

Problem 3 (pr3.c)

- Write a program to take an integer n from the user and print all prime numbers from 1 to n .
 - ✓ A *prime number* is a positive integer greater than 1 that has no positive integer divisors other than 1 and itself.
 - ✓ **Implement** a separate function to determine if a number is prime.
 - ✓ The prime-checking function should return 1 if the number is prime, and 0 otherwise.
 - ✓ Do not print 1 as it is not a prime number.
 - ✓ The function's prototype is:
`int isPrime(int num);`
 - ✓ If you program without implementing a separate function, -6 points will be deducted.

Problem 3 (pr3.c)

- **Program output**

```
[ohyong@cse Quiz1_s12]$ vi pr3.c
[ohyong@cse Quiz1_s12]$ gcc pr3.c -o pr3 -lm -std=c99
[ohyong@cse Quiz1_s12]$ ./pr3
Enter a positive integer: 8
Prime numbers from 1 to 8:
2
3
5
7
```

```
[ohyong@cse Quiz1_s12]$ ./pr3
Enter a positive integer: 24
Prime numbers from 1 to 24:
2
3
5
7
11
13
17
19
23
```

Submission

- **Submit to CSE server**

At the end of the Quiz1, submit your C source files by typing

```
~gs1401/bin/submit Quiz1_s12 pr1.c pr2.c pr3.c // due : 12:00
```

You may check that you have submitted your source code correctly by typing

```
~gs1401/bin/submit -check
```