Shiv Nadar University

CSD101: Introduction to Computing and Programming

Quiz #1

`Max marks: 35`

1. (5,5,5=15 points) Answer the following three questions with respect to the VSC program given below.
   (a) What will be printed out if $m = -10$, $n = -20$?
   (b) What is the code from address 15 to 20 doing? Give a short one line answer.
   (c) What is the entire program doing? Give a short one line answer.

(5,5,5=15 points)

Solution:

(a) -20
-10

(b) The code is exchanging the contents of addresses 100 and 101.

(c) The program reads two numbers and prints them in ascending order one number per line.
Note: The code does this by printing the contents of 100 first and then 101 so if $m > n$ it exchanges the contents of addresses 100 and 101 and then jumps to the printing part.

2. (2,3,2,4,3,3,3=20 points) The prime factorization of a positive integer $n > 1$ is the sequence of prime number divisors that when multiplied together yield n. Some examples of factorization are given below:

Value of n	Factorization
24	2 2 2 3
60	2 2 3 5
13	13

The C program fragment given below reads a positive integer $n > 1$ and prints out the prime factorization of n on the terminal. There are some missing parts in the program indicated by `??1??` to `??7??`. Fill in those parts suitably so that the program works correctly. Assume that the function **int isPrime(int n);** is available and can be called in the code fragment.

```c
int isPrime(int n);//returns True if n is a prime else False

int nextPrime(int m) {
  //Returns the smallest prime larger than m
  do  ??1??; while ??2??;
  return m;
}

int main() {
  int n,factor;
  printf("Give the +ve integer to be factorized = ");
  scanf(??3??);//Reads n
  factor=2;
  while (??4??) {
    if (??5??) {
      printf("%d ",factor);
      n=??6??;
    }
    else factor=??7??;
  }
  printf("\n");
  exit(0);
}
```

(2,3,2,4,3,3,3=20 points)

Solution:

Multiple solutions are possible.

1. `m++` or equivalent like `m=m+1`, `m+=1`.
2. `!isPrime(m)`.
3. `"%d",&n`.
4. `n>=factor` or `n>1`.
5. `n%factor==0`.
6. `n/=factor` or `n=n/factor`.
7. `factor=nextPrime(factor)`

There is an alternate solution given below without using `nextPrime`. However, it is less efficient and in any case you have to fill in `??1??` and `??2??` for the `nextPrime` function.

1. `m++` or equivalent like `m=m+1`, `m+=1`.

2. `!isPrime(m)`.

3. `"%d",&n`.

4. `n>=factor` or `n>1`.

5. `isPrime(factor) && n%factor==0`.

6. `n/=factor` or `n=n/factor`.

7. `factor=factor1+` or `factor++`