**German University in Cairo**
**Media Engineering and Technology**
**Assoc. Prof. Mervat Abuelkheir**
**Eng. Nada Ibrahim**
**Eng. Kamilia Hosny**
**Eng. Marina Nader**
**Eng. Nada Labib**
**Eng. Hager Khaled**

**Software Engineering**, Spring 2024
**Practice Assignment 8**

## Exercise 8-1

### Testing (Verification & Validation)

a) What is meant with the term "Testing" and what are the most important objectives thereof?

b) Explain software validation and software verification (V&V).

c) What are the activities involved in static inspection?

d) What are the activities involved in dynamic testing?

e) Why is inspection an important part of testing?

## Exercise 8-2

### Black box vs White box testing

Explain briefly the difference between black box and white box testing.

## Exercise 8-3

What is code coverage?

## Exercise 8-4

### Control Flow Graph

a) Provide a control flow graph for the following code.

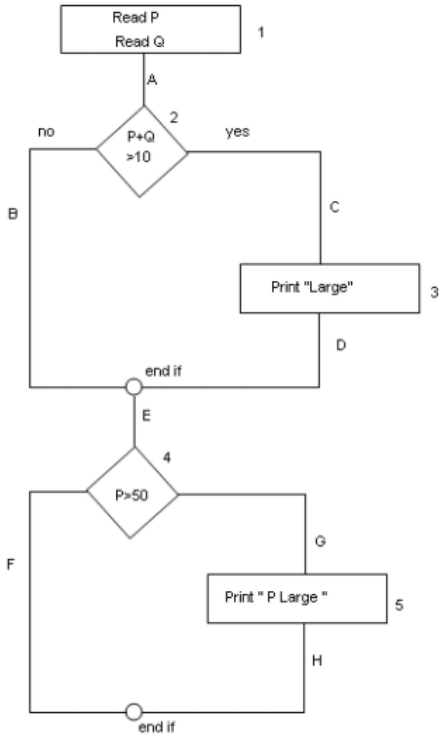
```
1 int myMethod()
2 {
3  int x, y;
4  cin >> x >> y;
5  While (x > 10)
6  {
7     x = x - 10;
8     if (x == 10)
9     break;
10    }
11    if ((y == 20) && ((x % 2) == 0))
12       y = y + 20;
13    else
14       y = y - 20;
15    return 2 * x + y;
16 }
```

b) Provide the path on the CFG given the following inputs: x=20 and y= 10.

c) Provide the path on the CFG given the following inputs: x=30 and y= 20.

**Exercise 8-5**

**White Box Testing Example**




Consider the following CFG.

a) Provide a pseudo code reflecting the functionality of the CFG.

b) Provide one path on the CFG that will have a full statement coverage.

c) Provide the paths that will cover all branches.

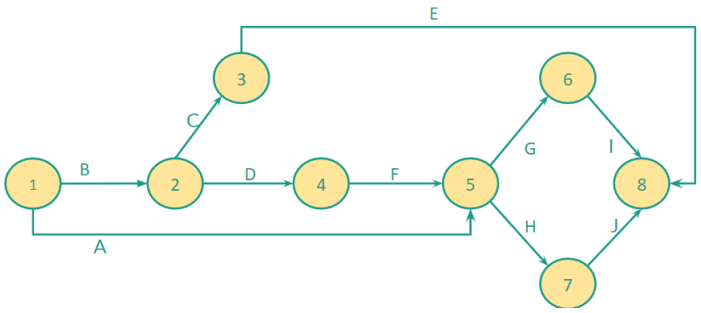
d) Provide all the possible paths from start to end in this flowchart.

e) What is the statement coverage, branch coverage and path coverage (%) for the following route: 1A, 2B, E, 4F?

f) What is the maximum coverage path?

**Exercise 8-6**

**White Box Testing Example**




Consider the following CFG.

- How many statements, branches and paths are there?
- Assume a test case that covers the following route: 1B,2D,4F,5G,6I,8. Calculate the statement coverage, branch coverage and path coverage (%).

**Exercise 8-7**

**Black Box Testing Example**

Consider the following requirements and given data.

```
/* Business specifications of the system
  A. Usernames must be unique
  B. All fields must only consist of charachters
  C. The maximum size of a username is 8 charachters
  D. All fields are required and must not contain spaces
  E. Password can not be less than 3 charachters
*/
const users = [
    {
        firstName: 'Tony',
        lastName: 'Stark',
        username: 'edith',
        password: 'jarvis'
    },
    {
        firstName: 'Stephen',
        lastName: 'Strange',
        username: 'wizzard',
        password: 'durmamo'
    },
    {
        firstName: 'Steve',
        lastName: 'Rogers',
        username: 'cap',
        password: 'peggy'
    },
    {
        firstName: 'Bruce',
        lastName: 'Banner',
        username: 'hulk',
        password: 'smash'
    },
]
```

For each of the following test cases, explain what they are exactly testing.

```
// Login with a username that does not exist
test('Login with username: vision should not be successful',() => {
  expect(login('vision','wanda')).toBe('User does not exist')
})

// Login with wrong password
test('Login with password: thanos for Tony should not be successful',() => {
  expect(login('edith','thanos')).toBe('Incorrect password')
})

// Successful case
test('Login with username: edith & password: jarvis should be successful',() => {
  expect(login('edith','jarvis')).toBe('Logged in successfully')
})

// Constraint A test 1
// Signup with an existing username
test('Sign up with username: edith should fail',() => {
  const newUser = {
    'firstName': 'Peter',
    'lastName': 'Parker',
    'username': 'edith',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('Username already exists')
})

// Constraint B test 1
// Signup with fields that contain numbers
test('Sign up with firstName: Peter1 should fail',() => {
  const newUser = {
    'firstName': 'Peter12',
    'lastName': 'Parker3',
    'username': 'edith',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('Can not have numbers in firstName')
})
```

```
// Constraint C test 1
// Signup with a username's length greater than 8
test('Sign up with username: thisIsAVeryLongUsername should fail',() => {
  const newUser = {
    'firstName': 'Peter',
    'lastName': 'Parker',
    'username': 'thisIsAVeryLongUsername',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('Max length for username is 8 charachters')
})


// Constraint D test 1
// Signup with some fields missing
test('Sign up without firstName should fail',() => {
  const newUser = {
    'lastName': 'Parker',
    'username': 'spider',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('Missing filed: firstName')
})

// Constraint D test 2
// Signup with a field containing spaces
test('Sign up with firstName containing spaces should fail',() => {
  const newUser = {
    'firstName': '    ',
    'lastName': 'Parker',
    'username': 'spider',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('Can not have spaces in firstName')
})
```

```
// Constraint E test 1
// Signup with a password's length less than three
test('Sign up with password: ma should fail',() => {
  const newUser = {
    'firstName': 'Peter',
    'lastName': 'Parker',
    'username': 'spider',
    'password': 'ma'
  }
  expect(signup(newUser)).toBe('Min length for password is 3 charachters')
})


// Successful case
// Observe all contraints
test('Sign up with username: Peter should succeed',() => {
  const newUser = {
    'firstName': 'Peter',
    'lastName': 'Parker',
    'username': 'spider',
    'password': 'may'
  }
  expect(signup(newUser)).toBe('User created successfully')
})
```

**Exercise 8-8**

**Black Box Testing Example**

Assume we have a function that given a student id and a course id, registers the student for the course. If we have a catalogue of all courses and students. what are the test cases that we can generate and what should we expect?