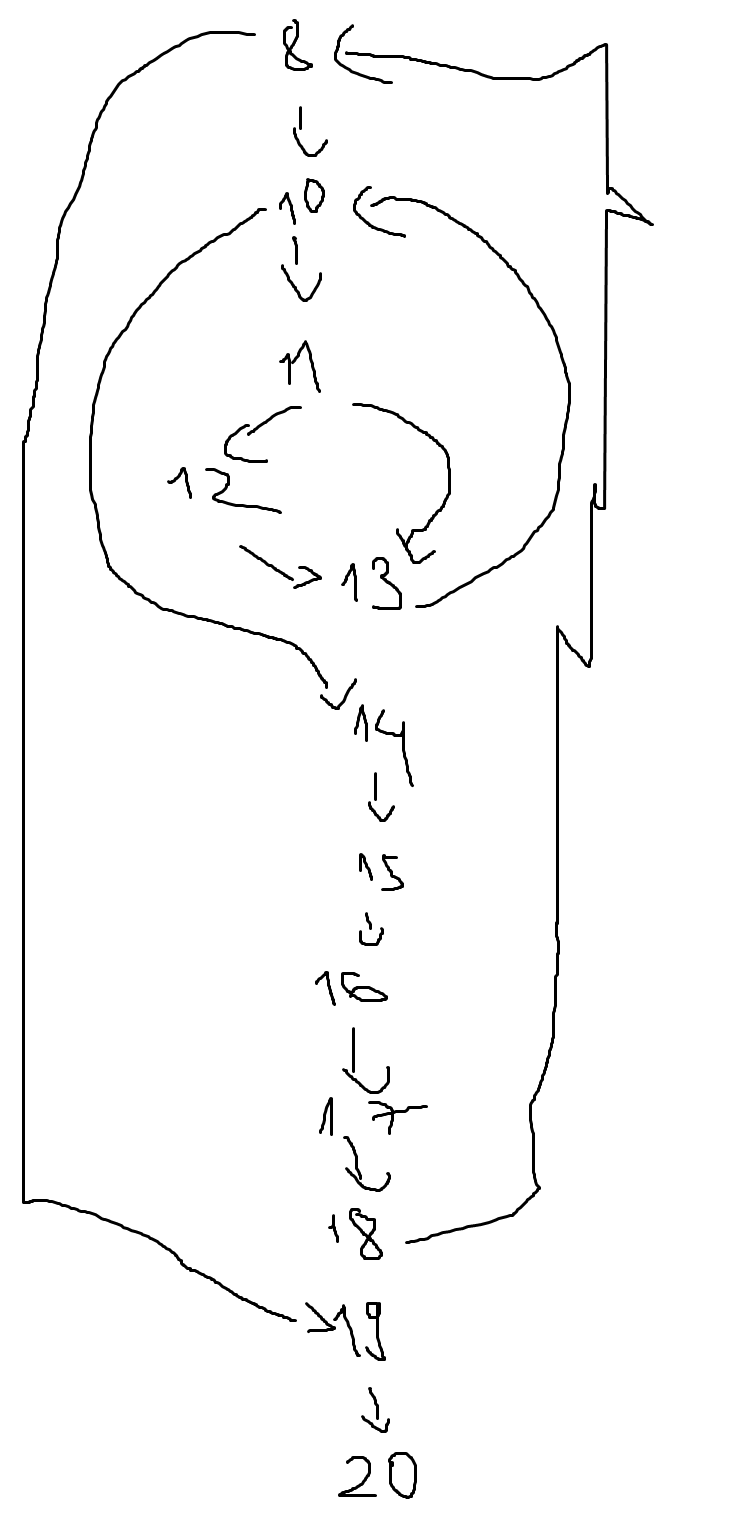
Câu 1.

a) Ta có Control-flow graph sau.



Số các node trong đồ thị là: N = 12

Số các cạnh là là: E = 14

Các đường thực thi suy ra được là: P = E – N + 2 = 4

Đường cơ sở

* 8->10->11->12->13->10->14->15->16->17->18->8->19->20.

Quy ước 14 -> 15-> 16-> 17 là 14 đến 17 , 18->8->19->20 là 18 đến 20.

Flip decision tại 11

8->10 -> 11-> 13🡪10->14 đến 17-> 18 đến 20.

Flip decision tại 10

8->10->14 đến 17 -> 18 đến 20.

Flip decision tại 8

8->19->20.

b)

Test-case detail:

* Path: basic path.
* Input: k = 1; size = 3 ; arr[] = {1,2,3}
* Expected output: 3.

\*\*Nháp

Để chạy đc vào vòng for đầu tiên thì k > 0.

Để vào được vòng for thứ 2 thì size > 1.

Để vào được câu lệnh if thì arr[max] < arr[i] , hay arr[0] < arr[1]

\*\*

Test-case detail:

* Path: Flip decision tại 11.
* Input: k = 1; size = 3; arr[] = {9,7,15}
* Expected: 15

\*\*Nháp

Flip decision tại 11, tức là không đi vào vòng if. -> arr[max] > arr[i] hay arr[0] > arr[1]

Do vào đc 2 vòng for -> k > 0 và size > 1.   
\*\*

Test-case detail:

* Path: Flip decision tại 10
* Input: k = 1; size = 1; arr[] = {69}
* Expected: 69

\*\*Nháp  
Flip decision tại 10 , tức là không đi vào vòng for hay size <= i. = 1

\*\*

Test-case detail:

* Path: Flip decision tại 8
* Input: k = 0; size = 3; arr = {1,2,3}
* Expected: 3.

\*\*Nháp:

Ko đi vào 8 -> k = 0

\*\*

Câu 2.

<https://t4tutorials.com/what-is-simple-boundary-value-testing-on-3-variables-software-testing/>

<https://swen90006.github.io/notes/Boundary-Value.html>

Ta nhận thấy input của chúng ta là biến marks[] và credits[] là 2 mảng. Biến num chỉ chứa một giá trị.

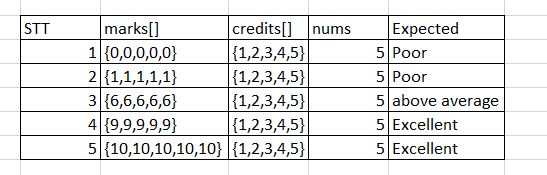
Xét biến marks, ta có .

* Giới hạn 0 <= marks <= 10 . Giá trị trung vị là 6
* Các giá trị đặc biệt của marks là 12 , 13 và 17

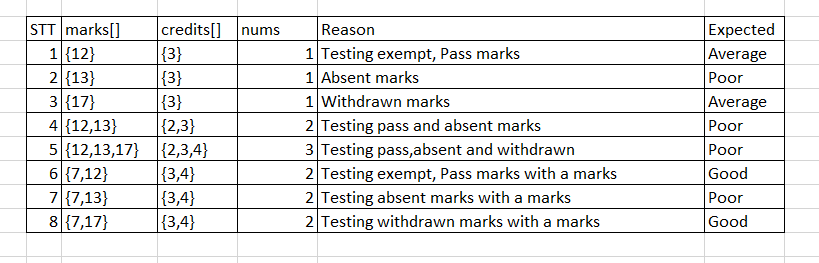
Xét biến credits, ta có: 1 <= credits <= 5. Giá trị trung vị là 3

Xét biến num , ta có num >= 0 .

Áp dụng kiểm thử giá trị biên xấu nhất, kết hợp với mảng, ta có các testcase sau.



Áp dụng kĩ thuật kiểm thử các giá trị đặc biệt, ta có



Áp dụng kĩ thuật Equivalence class partitioning, ta có

Ta có

\*Marks

- M1 = {mark: mark is 0 <= mark <= 10}

- M2 = {mark: mark is 12 (exempt, pass) }

- M3 = {mark: mark is 13 (absent, equal to 0.0)}

- M4 = {mark: mark is 17 (withdrawn)}

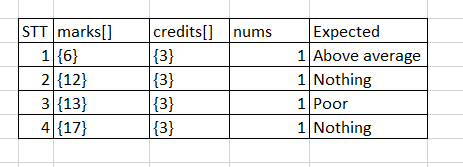
\*credits

- C1 = {credit: credit is 1 <= credits <= 6}

\*nums

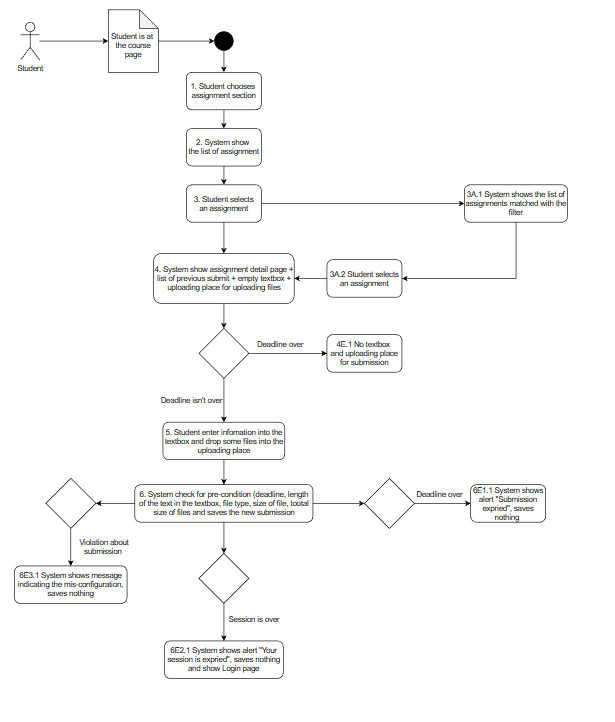
- nums1 = {num: num is greater or equal to zero}

Ta có bảng testcase sau.



Bài 3. Bài này, ta sẽ áp dụng Use-case testing.

Đầu tiên, ta sẽ vẽ Activity Diagram của các use-case.



Dựa vào diagram trên, ta có các đường thực thi sau.

* 1->2->3->4->5->6
* 1->2->3->3A.1->3A.2->4->5->6
* 1->2->3->4->4E.1
* 1->2->3->4->5->6->6E1.1
* 1->2->3->4->5->6->6E2.1
* 1->2->3->4->5->6->6E3.1

Dựa vào các đường thực thi, ta có các test scenario tương ứng.

Viết tắt: TS = Test scenario ; TN = Test name: PreC = Precondition ; PostC = Post condition ;

TS: Verify the basic flow.

TN: Basic flow

PreC: Deadline is not over and student’s assignment meets the system requirement.

PostC: Student successfully submitted assignment.

Test step: 1-> 2->3->4->5->6

Test data: Student’s assignment (files, text, … )

Test scenario: Verify if search for course works.

Test name: Alternative flow 1

Pre-condition: Deadline is not over and student’s assignment meets the system requirement.

Post condition: Student successfully submitted assignment.

Test step: 1-> 2->3->4->5->6

Test data:

* Student’s assignment (files, text, … )
* Search query: CO3015

Test scenario: Check if system know deadline is over 1.

Test name: Exceptions 1.

Pre-condition: Student can’t enter submit assignment site because deadline is over. Student’s assignment meets the system requirement.

Post condition: Student can’t submit their assignment.

Test step: 1-> 2->3->4->4E.1

Test data: Student’s assignment (files, text, … )

Test scenario: Check if system know deadline is over 2.

Test name: Exceptions 2.

Pre-condition: Student can enter the submit assignment site, but can’t submit because deadline is over. Student’s assignment meets the system requirement.

Post condition: System raises alert “Submission expired” , saves nothing.

Test step: 1-> 2->3->4->5->6->6E1.1

Test data: Student’s assignment (files, text, … )

Test scenario: Verify if student can submit assignment when session expired.

Test name: Exception 3.

Pre-condition: Student can’t enter the submit assignment site because he/she session is expired. Student’s assignment meets the system requirement.

Post condition: System raises alert “Your session is expired”, saves nothing and show the login page.

Test step: 1-> 2->3->4->5->6->6E1.2

Test data: Student’s assignment (files, text, … )

Test scenario: Verify if system can block violated submission

Test name: Exception 3.

Pre-condition: Deadline is not over. Student’s assignment doesn’t meet the system requirement.

Post condition: System saves nothing.

Test step: 1-> 2->3->4->5->6->6E1.3

Test data: Student’s assignment (files, text, … )