**Laboratory Exercise 6**

**Array and Pointer**

Họ và tên: Đỗ Mạnh Phương

MSSV: 20225660

Assignment 1.

Create a new project to implement procedure in Home Assignment 1. Add code the main program and initialize data for the integer list. Compile and upload to simulator. Run this program step by step, observe the process of explore each element of the integer list using indexing method.

A screenshot of a computer

Description automatically generated

A white background with green text

Description automatically generated

A close-up of a white page

Description automatically generated

A white background with green text

Description automatically generated

Giải thích cơ chế chương trình:

- Đầu tiên, ta cần tạo ra một size bất kỳ cho mảng ban đầu do khi nếu muốn thay đổi kích thước của mảng cần có giá trị tạm thời trước.

- Tiếp theo, ta sẽ tạo tin nhắn thông báo cho người dùng nhập giá trị của kích thước mảng )trong khoảng từ 0 đến 1000). Sau đó, người dùng sẽ nhập từ bàn phím giá trị mình mong muốn, hoàn thành bước khởi tạo mảng rỗng với kích thước do người dùng quyết định ở phần initiate\_the\_array.

- Ở hàm value\_info, ta sẽ đưa thông tin cho người dùng về việc tạo giá trị cho từ phần tử của mảng, với số thứ tự tương ứng với phần tử ta đang truy cập

- Ở hàm value\_prep, ta lưu mảng vào thanh ghi $t0, lưu kích thước thực của mảng vào thanh ghi $t1, lưu số thứ tự hiện tại của phần tử vào thanh ghi $t2.

- Ở hàm value, ta sẽ lần lượt nhập giá trị vào phần tử của mảng, bằng việc nhập giá trị cần tìm vào thanh ghi $v0, sau đó lưu giá trị này vào phần tử A[i] hiện tại bằng cách dùng lệnh sw $v0, 0($t0). Khi thực hiện việc lưu giá trị xong, ta tăng giá trị $t0 lên 4 để dùng tới con trỏ trỏ đến phần tử tiếp theo của phần tử A[i+1], đồng thời tăng giá trị của $t2 lên 1 để lấy số thứ tự tiếp theo của phần tử mà ta đang nhập. Lặp lại việc này đến khi ta nhập đủ các giá trị cho các phần tử. Khi đó, nếu giá trị thanh ghi $t2 = $t1, ta sẽ thực hiện lệnh rẽ nhánh để bắt đầu vào hàm main.

- Ở hàm main, ta sẽ khởi tạo, lưu mảng và kích thước của mảng vào thanh ghi $a0 và thanh ghi $a1, và nhảy đến hàm chính. - Ở hàm mspfx, ta lần lượt khởi tạo các giá trị như độ dài của mảng con cho thanh ghi $v0, giá trị tổng (tạm thời lớn nhất) cho thanh ghi $v1, chỉ số (vị trí của phần tử trong mảng) cho thanh ghi $t0, giá trị tổng hiện tại của mảng con cho thanh ghi $t1.

- Tiếp đến phần loop, do chương trình dùng chỉ số để chạy các phần tử của mảng, nên ta khởi tạo giá trị 4i, rồi cộng với giá trị thanh ghi $a0 để có được địa chỉ của phần tử A[i], lưu vào thanh ghi $t3. Tiếp tục, ta lưu giá trị của phần tử A[i] vào thanh ghi $t4. Sau đó, ta lấy giá trị của phần từ này cộng với tổng hiện tại ở thanh ghi $t1. Nếu giá trị của thanh ghi $v1 < $t1, giá trị của thanh ghi $t5 sẽ trả về 1, ngược lại là 0. Nếu $v1 < $t1, ta thực hiện việc rẽ nhánh , và thực hiện việc thay đổi độ dài của mảng con, đồng thời thay giá trị của tổng lớn nhất bằng tổng hiện tại, sau đó thực hiện việc tăng chỉ số I lên khi chỉ số này vẫn nhỏ hơn số lượng phần tử cho trước. Ngược lại, nếu $v1 > $t1, sẽ bỏ qua bước rẽ nhánh và nhảy lên tăng chỉ số

- Khi đã tính tổng hết các phần tử hiện tại, thì sẽ nhảy về hàm return để trả về kết quả.

=>Kết quả: Chương trình trên đã chạy đúng.

A screenshot of a computer

Description automatically generated

Assignment 2.

Create a new project to implement procedure in Home Assignment 2. Add code the main program and initialize data for the integer list. Compile and upload to simulator. Run this program step by step, observe the process of explore each element of the integer list using pointer updating method.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white paper with green text

Description automatically generated

A white screen with green text

Description automatically generated

=> Kết quả của chương trình:

A white background with black and white clouds

Description automatically generated



=>Như vậy, chương trình trên đã chạy đúng.

Assignment 3.

Write a procedure to implement bubble sort algorithm.

A white background with green text

Description automatically generated

A white background with green text

Description automatically generated

A screenshot of a computer

Description automatically generated

-Kết quả chạy chương trình:

A white background with black dots

Description automatically generated

A white background with a black border

Description automatically generated with medium confidence

Assignment 4.

Write a procedure to implement insertion sort algorithm.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with green text

Description automatically generated-Kết quả chạy chương trình:

Array after each round is:

-2 , 7 , 5 , 1 , 5 , 6 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 5 , 7 , 1 , 5 , 6 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 5 , 7 , 5 , 6 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 5 , 5 , 7 , 6 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 5 , 5 , 6 , 7 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 5 , 5 , 6 , 7 , 7 , 3 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 6 , 7 , 7 , 6 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 59 , 5

Array after each round is:

-2 , 1 , 3 , 5 , 5 , 5 , 6 , 6 , 7 , 7 , 8 , 8 , 59