**Laboratory Exercise 6**

**Array and Pointer**

**Đỗ Hải Dương – 20194528**

**Assignment 1**

**Graphical user interface, text, application

Description automatically generated**

**Text, letter

Description automatically generated**

* Kết quả chạy (max prefix sum lưu ở v1).

Max prefix sum = -2 + 6 -1 + 3 + 20194528 = 20194534

Table

Description automatically generated

**Giải thích:**

* Khai báo các giá trị của mảng
* Gán địa chỉ của A vào thanh ghi a0
* Gán a1 = 6 (vì mảng có 6 phần tử)
* Khởi tạo các giá trị v0 = 0, max prefix sum v1 = 0, t0 = i = 0, tổng chạy t1 = 0
* Vòng lặp (nhãn loop) gán t2 = 2i
* Tiếp theo gán t2 = 4i
* Gán địa chỉ A[i] vào thanh ghi t3 🡪 load giá trị của A[i] vào t4 🡪 thêm A[i] vào tổng chạy t1
* So sánh giá trị v1 (tổng cần tìm) với t1 (tổng cần chạy)

+) Nếu v1 < t1 thì t5 = 1

+) Nếu t5 != 0 thì nhảy đến nhãn mdfy

* Trong nhãn mdfy: dãy max prefix sum mới có độ dài i + 1
* Gán v1 = t1
* So sánh: Nếu t0 ( biến chạy) < a1 (số phần tử của mảng) thì t5 = 1
* Nếu t5 != 0 (tức i < n) thì quay lại vòng lặp loop

**Assignment 2**

**Text

Description automatically generated**

**A picture containing text

Description automatically generated**

**Kết quả:**



**Giải thích:**

* + Khai báo các giá trị của mảng
  + Khai báo Aend
  + Gán địa chỉ của A vào thanh ghi a0, gán địa chỉ của Aend vào thanh ghi a1
  + Giảm a1 đi 4 🡪 địa chỉ của A[n-1]
  + Nhãn after\_sort: gán v0 = 10 (10 là exit)
  + Nhãn sort: so sánh nếu a0 = a1 thì nhảy đến nhãn done
  + Nhãn after\_max: lưu giá trị của phần tử cuối vào t0
  + Copy giá trị phần tử cuối cùng vào vùng lớn nhất 🡪 copy giá trị lớn nhất vào phần tử cuối
  + Giảm địa chỉ đi 4 🡪 nhảy quay lại sort cho mảng bé hơn ( trừ phần tử đã đưa về cuối)
  + Nhãn done: nhảy đến after\_sort để thoát
  + Gán địa chỉ v0 = a0 (v0 là con trỏ lớn nhất) 🡪 load địa chỉ của v0 vào v1 🡪 gán t0 về địa chỉ đầu tiên của mảng
  + So sánh nếu phần tử tiếp theo là cuối cùng thì nhảy đến ret
  + Tăng t0 lên 4 🡪 địa chỉ của t0 (phần tử tiếp theo) vào t1
  + So sánh nếu t1 < v1 thì t2 = 1. Nếu t2 != 0 thì quay lại vòng lặp loop
  + Gán phần tử tiếp theo theo thành phần tử lớn nhất mới 🡪 gán giá trị t1 vào v1 (giá trị phần tử kế tiếp là giá trị max)

**Assignment 3**

Chart

Description automatically generated with low confidence

Table

Description automatically generated

Table

Description automatically generated with medium confidence

**Kết quả:**

Text

Description automatically generated

**Giải thích:**

* Khai báo mảng, khai báo xâu space(dấu cách để in mảng)
* Gán địa chỉ của mảng arr vào thanh ghi s0, gán biến chạy i = 0, gán s1 = 8 (phần tử mảng từ 0 đến 8), s2 =0 (số phần tử mảng) để giảm dần, t2 = địa chỉ của mảng
* Nhãn outer\_loop: gán t1 = 0 (biến j), giảm s2 đi 1, gán t3 = địa chỉ của arr[0]
* Nhãn inner\_loop: load giá trị của arr[j] vào s3, tăng t3 lên 4, load giá trị arr[j+1] vào s4, tăng t1 lên 1(biến j)
* So sánh nếu s3 < s4 thì t4 = 1

Nếu t4 != 0 thì nhảy đến cond

* Nhãn swap: ghi địa chỉ của s3 vào t3, ghi địa chỉ của t3 – 4 vào s4 🡪 ghi giá trị của t3 vào s4
* Nhãn cond: so sánh nếu t1 ! = s2 thì nhảy đến inner\_loop 🡪 tăng t0(biến i) lên 1
* So sánh nếu t0 != s1 thì nhảy đến outer\_loop
* Nhãn print: in mảng
* Nếu t0 != s1 (chưa đến cuối mảng) thì nhảy lại đến print để in phần tử còn lại.