**Assignment 1:**

* Mã nguồn:

Graphical user interface, text, application, email

Description automatically generated

Text, letter

Description automatically generated

* Màn hình chạy:

Graphical user interface, application, Excel

Description automatically generated

* Kết quả = 20194592 => Chính xác
* Giải thích:
* Dòng 1,2: Khai báo mảng
* Dòng 4: load địa chỉ A vào a0
* Dòng 5: Gán a1 = 6 ( = số phần tử)
* Dòng 6: Chạy mspfx
* Mspfx (dòng 24 – 35):
* Dòng 24-27: Khởi tạo các giá trị
* Dòng 28,30: Lấy từng phần tử của A
* Dòng 31: load phần tử hiện tại (t3) vào t4
* Dòng 32: This running sum
* Dòng 33-34: So sánh running sum với max sum, nếu max sum nhỏ hơn => thực hiện modify (Gán lại max sum dòng 36,37)
* Dòng 35:kiểm tra điều kiện lặp với hàm test
* Dòng 41: Sau khi xong thì lặp vô hạn để chờ tín hiệu tiếp

**Assignment 2:**

* Mã nguồn:

Text, application

Description automatically generated

Text

Description automatically generated

Text, letter

Description automatically generated

* Màn hình chạy:

A screenshot of a computer

Description automatically generated with medium confidence

* Giải thích:
* Dòng 1-3: Khai báo mảng
* Dòng 5-7: load địa chỉ của phần tử đầu, cuối của A vào a0,a1
* Dòng 8: Thực hiện hàm sort
* Dòng 9-11: Sau khi sort xong thì end chương trình
* Hàm sort (Dòng 21-22):Nếu còn 1 phần tử thôi thì nhảy đến hàm max
* Hàm done (Dòng 28): Nhảy đến after\_sort
* Hàm after\_sort: Thoát khỏi chương trình
* Hàm loop (Dòng 39-47):
* Dòng 40: So sánh next = last thì thoát
* Dòng 41,42: Lấy phần tử tiếp theo
* Dòng 42: So sánh next và max
* Dòng 43: Nếu < => tiếp tục loop
* Dòng 45,46: Cập nhận next và max mới
* Dòng 47: Tiếp tục loop

**Assignment 3:**

* Mã nguồn:

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

A picture containing calendar

Description automatically generated

* Màn hình chạy:

Graphical user interface, application

Description automatically generated

* Giải thích:
* Dòng 1-3: Khai báo mảng
* Dòng 7-8: load mảng vào a0 và gán a1 = số lượng phần tử
* Dòng 9: nhảy đến hàm sort
* Hàm sort( Dòng 16): Khởi tạo t0 (index i) = 0
* Chạy loop\_1 (Dòng 19): Khởi tạo index j của vòng lặp thứ 2, j++, t2 = n – j
* loop\_2 (Dòng 24): Lấy ra A[j], A[j+1]
* if (Dòng 30): Tiến hành so sánh A[j] và A[j+1], nếu bé hơn thì end\_if, còn > thì phải swap
* end\_if: Tăng j = i+j, so sánh với n-j để kiểm tra điều kiện lặp
* end\_loop\_2: So sánh i với n để kiểm tra điều kiện lặp

**Assignment 4:**

* Mã nguồn:

Text

Description automatically generated with medium confidence

Text

Description automatically generated

* Màn hình chạy:

A screenshot of a computer

Description automatically generated with medium confidence

* Giải thích:
* Dòng 2,3: Khởi tạo mảng
* Dòng 6,7: load địa chỉ của A vào a0 và khởi tạo độ dài của A vào a1
* Dòng 8: Nhảy đến hàm sort
* Hàm sort (Dòng 13-15): Khởi tạo i = 0
* Hàm loop (Dòng 17-23) : Lấy ra A[i] và cho j = i-1
* Hàm while (Dòng 25-44):
* Dòng 27,28: So sánh j với 0, nếu <0 thì end\_while
* Dòng 30-33: Lấy ra A[j+1]
* Dòng 35-42: So sánh A[j] và A[j+1], nếu > thì tiến hành end\_while
* đổi chỗ, giảm j
* End\_while (dòng 45):
* Tăng j = j+1
* Lấy ra A[j+1] và A[j] để đổi chỗ
* Kiểm tra điều kiện I <n để end\_loop