Họ & tên  -   MSSV

Nguyễn Hải Đăng - 1913094

Trần Quyết Thắng - 1915251

Vũ Minh Dương - 1810885

Solutions:

1.

static: mảng khởi tạo trên vùng nhớ static, kích thước mảng xác định tại thời điểm compile, truy cập nhanh và an toàn vào bộ nhớ

advantages: hiệu quả vì không cần cấp phát động

ex: static float arr[100];

fixed stack-dynamic: mảng khởi tạo trên vùng nhớ stack, kích thước mảng xác định tại thời điểm compile

advantages: hiệu quả về không gian, các mảng lớn ở những chương trình con khác nhau có thể chia sẻ chung không gian, ngay khi các chương trình con này không hoạt động ở cùng 1 thời điểm; hỗ trợ đệ quy

ex: float arr[100];

stack-dynamic: mảng khởi tạo trên vùng nhớ stack, kích thước mảng được xác định tại thời điểm chạy chương trình

advantages: khởi tạo trên stack nên vùng nhớ sẽ được tự cấp phát lại khi hết vòng đời, linh hoạt hơn so với static array và fixed stack-dynamic array

ex:

cin >> x;

float arr[x];

fixed heap-dynamic: mảng khởi tạo trên vùng nhớ heap, kích thước mảng xác định tại thời điểm compile

advantages: linh hoạt, kích thước mảng phù hợp với vấn đề.

ex: int[] arr = new int[100];

heap-dynamic:  mảng khởi tạo trên vùng nhớ heap, kích thước mảng được xác định tại thời điểm chạy chương trình

advantages: sử dụng vùng nhớ heap cho phép có kích cỡ lớn hơn; linh hoạt hơn so với những dạng còn lại vì nó có thể được giải phóng và cấp phát lại tại thời điểm chạy chương trình

ex:

cin >> n;

int[] arr = new int[n];

2.

a.

b.

1.25

1 = 001

0.25 = 0.01000

1.25 = 1.01000

—----------------------------------------------------------------------

Đặng Hoài Bão - 1912697

Cao Thanh Bình - 1912715

Nguyễn Hữu Anh Đại - 2020020

1.

static: Bộ nhớ được cấp phát trên vùng nhớ static với kích thước cố định.

vd: khai báo mảng toàn cục hoặc được khai báo với từ khóa static.

static int x[10].

fixed\_stack: Bộ nhớ được cấp phát trên vùng stack với kích thước cố định và biết trước tại thời điểm compile

vd: khai báo mảng với kích thước cố định trong tầm vực cục bộ (scope)

int x[10]; //khai báo trong hàm

Ưu điểm:

* Hiệu quả không gian
* Hỗ trợ đệ quy

stack\_dynamic: được cấp phát trên vùng stack với kích thước không cố định tại thời điểm thực thi.

Ưu điểm:

* Tính linh hoạt

vd: khai báo mảng với kích thước là biến trong tầm vực cục bộ.

cin >> n;

int x[n];

fixed\_heap: được cấp phát trên vùng heap với kích thước cố định.

Ưu điểm:

* Tính linh hoạt
* Kích thước mảng luôn đáp ứng mọi vấn đề

vd: khai báo mảng với kích thước cố định sử dụng từ khóa yêu cầu cấp phát bộ nhớ động.

int[] x = new int[10];

heap-dynamic: Được cấp phát trên bộ nhớ heap với kích thước linh động tại thời điểm thực thi.

Ưu điểm:

* Kích thước của mảng có thể thu nhỏ hoặc mở rộng trong quá trình thực thi

vd: khai báo mảng với từ khóa yêu cầu cấp phát bộ nhớ kích thước từ biến

cin >> n;

int[] x = new int[n];

2.

-0.375\*2^1=-0.75

0.5\*2^2=2

0.5\*2^7=64

1. 1.25 → 0 011 01000

11.25 → 0 110 01101

111.25 → 0 111 10111

3.

2

+16+2(pad)

+4

+6 (4)

= 30

===========================================================

============================================================

Võ Nguyễn Thiện Nhân 1910409

Nguyễn Ngô Thanh Trúc 1910650

Phạm Thế Khang 1812558

Câu 1.

**Static array**: là 1 mảng đã biết số lượng phần tử tại thời điểm chương trình biên dịch

ex: static int x[10]

**Fixed stack-dynamic**: là 1 mảng mà kích thước mảng được biết tại thời điểm biên dịch, nhưng cho phép nó được phân bổ tự động trên vùng nhớ stack

ex: int x[10]

**Stack dynamic:** là 1 mảng mà kích thước của mảng chỉ được biết tại thời điểm chương trình đang chạy (runtime), mảng phân bổ trên vùng nhớ stack

ex: cin >> n

int x[n]

**Fixed heap-dynamic:** là 1 mảng mà kích thước mảng được biết tại thời điểm biên dịch, nhưng cho phép nó được phân bổ tự động trên vùng nhớ heap

int [] x = new int[10]

**Heap dynamic:** là 1 mảng mà kích thước của mảng chỉ được biết tại thời điểm chương trình đang chạy (runtime), mảng phân bổ trên vùng nhớ heap

cin >> n

int [] x = new int[n]

Câu 2.

a)

1 001 01100:

sign bit = 1 → negative

001: 1 (dec) → bias adjustment: 1 - (2^2 - 1) = -2

01100 →  - (1 + 0.2^0 + 1.2^-1 + 1.2^-2) = -1.75

Represent: -1.75 . 2^-2 = -0.4375

0 010 10000

sign bit =  → non-negative

010: 2 (dec) → bias adjustment: 2 - (2^2 - 1) = -1

10000 →  1.2^0 = 1

Represent: -1.75 . 2^-2

b)

**1.25**

sign bit = 0

phần nguyên: 1 = 1

phần thập phân:  01

1.25 = 1.01 \* 2^0

phần mũ: 0 + 3 = 3

=> IEEE754: 0 011 01000

**11.25**

sign bit = 0

phần nguyên: 11 = 1011

phần thập phân: 01

11.25 = 1011.01 = 1.01101 \* 2^3

phần mũ: 3 + 3 = 6

=> IEEE754: 0 110 01101

**111.25**

sign bit = 0

phần nguyên: 11 = 1111111

phần thập phân:  01

111.25 = 1111111.01 = 1.01111001 \* 2^6

phần mũ: 3 + 6 = 9

=> IEEE754: 0 111

Question 3

2 + 14 + 16 + 4 + 1 + 3 + 4 + 2 + 2

============================================================

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Họ và tên - MSSV:

Nguyễn Văn Quang - 1914802

Trần Tiến Lộc - 1914038

Bùi Xuân Thông - 1915350

Solutions:

Ques 1:

* Static:
* fixed stack-dynamic:
* stack-dynamic:
* fixed heap-dynamic:
* heap dynamic:
* Advantage:
* static, heap và stack-dynamic: bộ nhớ được cấp phát trong quá trình thực thi nên có tính cơ động, linh hoạt hơn
* fixed stack-dynamic và fixed heap-dynamic: bộ nhớ được cấp phát trong quá trình biên dịch nên có thể tính toán được kích thước bộ nhớ tối đa

Ques 2:

a.

1 001 01100: -1.375

0 010 10000:  3.5

0 111 10000:  7.5

    b.  
 1.25  : 000101000

11.25: 0 011 01000

111.25: không biểu diễn được vì phần exponent chỉ có 3 bits nên lớn nhất 7

    Ques 3:

==============================

Nguyễn Thọ Nam - 1911650

Lê Hồng Phong - 1813518

Lê Hoàng Anh - 1910752

Question 1:

* Static array là một mảng trong đó các phạm vi chỉ số con được ràng buộc tĩnh và cấp phát bộ nhớ tĩnh (được hoàn thành trước thời gian chạy).
* Fixed stack-dynamic array là một mảng trong đó các phạm vi chỉ số con được ràng buộc tĩnh, nhưng quá trình cấp phát bộ nhớ được thực hiện trong thời gian thực thi.
* Stack-dynamic arrays là một mảng mà trong đó cả phạm vi chỉ số con lẫn cấp phát bộ nhớ đều là động tại thời điểm thực thi. Size của array chỉ được xác định khi array được gọi sử dụng.
* Fixed heap-dynamic array là một mảng mà trong đó phạm vi chỉ số con được lẫn cấp phát bộ nhớ đều là động, nhưng cả hai đều trở nên cố định sau khi được cấp phát bộ nhớ.
* Heap-dynamic array là một mảng mà trong đó phạm vi chỉ số con được lẫn cấp phát bộ nhớ đều là động, và cả hai đều có thể thay đổi trong suốt thời gian tồn tại của mảng.

Question 2:

1. Convert binary floating point to decimal number:

* 100101100:

bit đầu: 1 → số âm

3 bit sau: 001 → số 1 = 2^x/2-1,( x là số bit )+ mũ -> mũ = -2

==============================

Trương Vĩnh Phước - 1910473

Huỳnh Thành Đạt - 1910110

Trần Quang Huy - 1812426

Question 1:

1)

* static array: là array được tạo tại compile time, là biến toàn cục, tồn tại theo dòng đời của chương trình. Ví dụ: static inat[3];
* fixed static-dynamic array: array được tạo tại runtime, có size cố định, tồn tại theo dòng đời của hàm nó thuộc. Ví dụ: int a[4];

2)

a)

1| 001 | 01100

sign = 1 → âm

f = 0.01100 = 0.375

e = 001 - 011 = -2

→ - (1+ 0.375) \* 2^(-2) = -0.34375

0| 010 | 10000

sign = 0 → dương

f = 0.10000 = 0.5

e = 010 - 011 = -1

→ (1 + 0.5) \* 2^(-1) = 0.25

0 | 111 | 10000

sign = 0 → dương

f = 0.10000 = 0.5

e = 111 - 011 = 4

→ (1 + 0.5) \* 2^4 = 24

b)

- 1.25

sign = 0

1.25 = 1.01 \* 2^0

→ f = 01, e = 0 + 011 = 011

0 | 011 | 01000

11.25

sign = 0

11.25 = 1011.01 = 1.01101 \* 2^3

→ f = 01101, e = 011 + 011 = 110

0 | 110 | 01101

111.25

sign = 0

111.25 = 1101111.01 = 1.10111101 \* 2^6

→ f = 10111, e = 110 + 011 = 001

0 | 001 | 01101

3)

A: 2 bytes

→ padding 14 bytes

B : 16 bytes

C: 4 bytes