

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

Đề tài: Khảo sát số chấm động

Môn học: CSC10009 - Hệ thống máy tính



Sinh viên thực hiện:

Nguyễn Quang Thái (23127116)

Giáo viên hướng dẫn:

Lê Việt Long

Ngày 31 tháng 10 năm 2024

Mục lục

1	Đánh giá	1
2	Kết quả bài làm	1

1 Đánh giá

Các yêu cầu	Tình trạng
Nhập vào số chấm động biểu diễn nhị phân của số chấm động vừa nhập	Hoàn thành
Nhập vào biểu diễn nhị phân của số chấm động xuất ra biểu diễn thập phân tương ứng	Hoàn thành
Trả lời các câu hỏi yêu cầu của bài tập 3	Hoàn thành
Viết chương trình và chạy và trả lời các yêu cầu của bài tập 4	Hoàn thành
Tổng thể	100 % Hoàn Thành

Đánh giá tổng thể:

Bài tập 1: Hàm `dumpFloat()` được xây dựng, hoàn thành tốt .

Bài tập 2: Hàm `forceFloat` được xây dựng, hoàn thành tốt.

Bài tập 3: Đã trả lời được các câu hỏi.

Bài tập 4: Đã viết chương trình chạy thử và hoàn thành bài tập.

2 Kết quả bài làm

```
$ g++ Ex1.cpp && ./a
Enter a float number: 3.7
0 10000000 11011001100110011001101
```

Hình 1: Bài tập 1

```
$ g++ ex2.cpp && ./a
Binary: 01000100001101100001000000000000:
Float:728.25
Binary: 10100011001101101100000000000000:
Float:-9.83913e-018
Binary: 00111101110011001100110011001101:
Float:0.1
Binary: 01111111100000000000000000000000:
Float:inf
Binary: 01111111100000000000000000000000:
Float:nan
```

Hình 2: Bài tập 2

Bài tập 3: $1.3E+20$ có biểu diễn nhị phân là: '0 11000001 11000011000001110011001'

Số float nhỏ nhất lớn hơn 0 là: '1.4013e-045'

Biểu diễn nhị phân là: '0 00000000 000000000000000000000001'

Những trường hợp tạo ra số đặc biệt kiểu float:

- Số vô cùng ('inf'): 'X / 0', '+ ∞ - ∞ ', ...

Là các số có giá trị vượt ngoài khả năng biểu diễn của hệ thống.

- Số báo lỗi ('NaN'): '0 / 0', 'sqrtx' với 'x < 0', ...

Khi các số nhập vào là số lỗi, vi phạm các nguyên tắc thông thường,...

<pre>\$ g++ Ex3.cpp && ./a Float: 1.3e+020 Binary: 0 11000001 11000011000001110011001</pre>	<pre>\$ g++ Ex3.cpp && ./a Float: 1.4013e-045 Binary: 0 00000000 000000000000000000000001</pre>
-----------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------

Hình 3: Bài tập 3

Bài tập 4: 1. Chuyển đổi 'float -> int -> float'. Kết quả như ban đầu?

Trả lời: Không, vì khi chuyển từ 'float' sang 'int', các số ở phần thập phân bị loại bỏ.

2. Chuyển đổi 'int -> float -> int'. Kết quả như ban đầu?

Trả lời: Có, vì từ 'int' sang 'float' và ngược lại không làm thay đổi giá trị.

3. Phép cộng số chấm động có tính kết hợp không? $(x + y) + z = x + (y + z)$

Trả lời: Có, kết quả của hai phép tính bằng nhau.

4. `i = (int)(3.14159 * f);`

Trả lời: 'i' sẽ lấy phần nguyên của kết quả phép nhân.

5. `f = f + (float)i;`

Trả lời: 'f' sẽ có giá trị mới là $f + i$.

6. `if (i == (int)((float) i)) printf("true");`

Trả lời: 'true'. Vì 'int -> float -> int' không làm thay đổi giá trị.

7. `if (i == (int)((double) i)) printf("true");`

Trả lời: 'true'. Việc chuyển 'int -> double -> int' cũng không thay đổi giá trị.

8. `if (f == (float)((int) f)) printf("true");`

Trả lời: 'false'. Việc chuyển 'float -> int -> float' làm mất phần thập phân.

9. `if (f == (double)((int) f)) printf("true");`

Trả lời: 'false', do 'float -> int -> double' thay đổi giá trị.

```
Nguyễn Quang Thái@MSI MINGW64 /d/HCMUS/KI_4/HTMT/Ex2
• $ g++ Ex4.cpp && ./a
1:
float = 3.7
float -> int = 3
float -> int -> float = 3
2:
int = 5
int -> float = 5
int -> float -> int = 5
3:
(x+y)+z == x+(y+z)
Co tinh ket hop
4:
i = 7
5:
f = 3.5
6:
true
7:
true
8:
false
9:
false
```

Hình 4: Bài tập 4