Câu 1:

**#include<bits/stdc++.h>**

**using namespace std;**

**typedef union {**

**float f;**

**struct**

**{**

**unsigned int M : 23;**

**unsigned int E : 8; //số mũ**

**unsigned int S : 1; // dấu +(0) hoặc trừ(1)**

**} raw;**

**} myfloat;**

**void OutBinary(int n, int i) {**

**int k;**

**for (k = i - 1; k >= 0; k--) {**

**if ((n >> k) & 1) {cout << "1";}**

**else {cout << "0";}**

**}**

**}**

**void dumFloat(myfloat x){**

**cout << x.raw.S << " | ";**

**OutBinary(x.raw.E, 8);**

**cout << " | ";**

**OutBinary(x.raw.M, 23);**

**cout << endl;**

**}**

**int main() {**

**myfloat x;**

**cout << "Nhap so cham dong x: ";**

**cin >> x.f;**

**cout << "ket qua sang nhi phan la:\n";**

**dumFloat(x);**

**return 0;**

**}**

Kết quả:

A screenshot of a computer

Description automatically generated with medium confidence

Câu 2:

**#include <bits/stdc++.h>**

**using namespace std;**

**typedef union {**

**float f;**

**struct**

**{**

**unsigned int M : 23;**

**unsigned int E : 8;**

**unsigned int S : 1;**

**} raw;**

**} myfloat;**

**unsigned int forceFloat(unsigned int\* arr, int low, int high) {**

**unsigned int f = 0, i;**

**for (i = high; i >= low; i--){**

**f = f + arr[i] \* pow(2, high - i);**

**}**

**return f;**

**}**

**int main() {**

**string s;**

**cout << "Nhap day 2phan 32 bit : ";**

**getline(cin, s);**

**unsigned int IEEE[32]; int n = 0;**

**for (int i = 0; i < s.size(); i++) {**

**if (s[i] != ' '){**

**IEEE[n] = s[i] - '0';**

**n++;**

**}**

**}**

**myfloat x;**

**unsigned int f = forceFloat(IEEE, 9, 31);**

**x.raw.M = f;**

**f = forceFloat(IEEE, 1, 8);**

**x.raw.E = f;**

**x.raw.S = IEEE[0];**

**cout << "Ket qua sang thap phan la:\n";**

**cout << x.f << endl;**

**return 0;**

**}**

Câu 3:

- bằng cách gán trực tiếp **x.f** ở câu 1 trong hàm main = 1.3E+20, chương trình sẽ tự hiểu đó là số thuộc kiểu float và truyền tham số vố các hàm đã tạo và thực hiện chuyển đổi, kết quả là:

1.3E+20 sang nhị phân : 0 | 11000001 | 11000011000001110011001

- Để trả lời câu hỏi trên, tham khảo các tài liệu và một số nguồn code từ các trang cộng đồng, ta có đoạn code truy tìm như sau:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**float x = 0;**

**char\* p = (char\*)&x;**

**(\*p) |= 1;**

**cout << x << endl;**

**return 0; }**

Từ đoạn code trên, ta có kết quả là: 1.4013e-45

Sau đó gán số trên vào **x.f** ở chương trình câu 1 để chuyển đổi và được dãy số là:

0 | 00000000 | 00000000000000000000001

- Các trường hợp đặc biệt:

+ số báo lỗi NaN khi dấu là 0 và các số còn lại đều là 1, do mũ quá lớn và phần thập phân quá lớn

+ số vô cùng (inf) phần mũ đều là mũ, còn lại là 0

Câu 4:

.1 từ float sang int và chuyển ngược lại kết quả sẽ không như ban đầu, do khi ép kiểu sẽ chỉ lấy phần nguyên trước dấu chấm như dưới:

Text

Description automatically generated

.2 khi chuyển đổi từ int sang float và sang int, giá trị không đổi do số nguyên vẫn là số thực với phần thực sau dấu chấm là 0

.3 phép cộng dấu chấm động có tính kết hợp do kết quả phụ thuộc vào kiểu dữ liệu trả về chứ không phụ thuộc vào thứ tự cộng:

.4 .5 .6 .7 .8 .9 được đúc kết như sau:

#include <iostream>

using namespace std;

int main()

{

int i;

float f = 1.2;

i = (int)(3.14159 \* f);

cout << i<<endl;

if (i == (int)((float)i)) cout << "true\n";

if (i == (int)((double)i)) cout << "true\n";

if (f == (float)((int)f)) cout << "true\n";

if (f == (double)((int)f)) cout << "true\n";

}

=> Khi ép kiểu từ số thực sang số nguyên thì số sẽ thay đổi và chỉ lấy phần nguyên trước dấu phẩy. Ngược lại khi ép kiểu từ số nguyên sang số thực kết quả sẽ không thay đổi do về bản chất sô nguyên vẫn là số thực với các phần thập phân bằng 0.