

Podstawy programowania, Czwartek 30.04.2015, 16:15-17:45

Zadanie 8

Zadanie polega na napisaniu programu, wykorzystując operacje bitowe, który dla danej grupy krwi biorcy wyznaczy grupy krwi, które mogą posiadać potencjalni dawcy, aby można było bezpiecznie przetoczyć pacjentowi krew. W programie podajemy grupę krwi biorcy i na wyjściu otrzymujemy grupy krwi zgodne z grupą krwi pacjenta. Przykład działającego programu:

Podaj grupe krwi:

AB-

Grupa krwi: AB-

Grupy zgodne: O- B- A- AB-

Zakładamy, że grupy krwi są kodowane bitowo:

A+=101, AB-=110, O+=001, B-=010, itd. (samodzielnie wyprowadzić kodowanie bitowe wszystkich grup krwi)

Uwaga! Tabela zgodności grup krwi jest następująca:

Tabela zgodności grup krwi								
Biorca	Dawca							
	O-	O+	B-	B+	A-	A+	AB-	AB+
AB+	X	X	X	X	X	X	X	X
AB-	X		X		X		X	
A+	X	X			X	X		
A-	X				X			
B+	X	X	X	X				
B-	X		X					
O+	X	X						
O-	X							

Do wykonania:

1. Stwórz strukturę Pacjent zawierającą składniki: uint8_t grupaKrw, int liczbaZgodnychGrup, uint8_t *grupyZgodne. Umieść definicję struktury w pliku Pacjent.h. Stwórz 2 funkcje powiązane ze strukturą Pacjent: void UstawPacjent(Pacjent *pacjent, char *grKrw) oraz void WypiszPacjent(Pacjent *pacjent) ustawiające i wypisujące składniki struktury DaneOsobowe. Umieść deklaracje funkcji w pliku Pacjent.h a ich definicje w pliku Pacjent.c. Ustawiamy poleceniem typedef nazwę struktury na Pacjent. (2 p.)

Główna część zadania polega na wyznaczeniu grup krwi zgodnych z podaną grupą krwi pacjenta. Zadanie to należy wykonać przeprowadzając odpowiednie operacje bitowe w następujący sposób: grupę krwi pacjenta sprawdzamy (operacjami bitowymi) z każdą inną grupą krwi. W wyniku operacji powinniśmy dostać wartość „true” lub „false”, w zależności od tego, czy porównywane grupy krwi są zgodne, czy nie. Jeśli dana grupa krwi jest zgodna z grupą krwi pacjenta, to zwiększamy o 1 liczbę zgodnych grup krwi danego pacjenta (składowa `liczbaZgodnychGrup`) i dodajemy daną grupę krwi do tablicy `*grupyZgodne` struktury `Pacjent`. **(2.5 p.)**

Całość kompilujemy w linii poleceń używając pliku `Makefile` oraz polecenia `make`. **(0.5 p.)**

Zrozumienie, analiza treści, projekt programu + poprawność i estetyka kodu. **(1 p.)**

Uwaga!

W plikach `.h` (gdzie mamy definicje struktur) używamy następujących instrukcji preprocesora:

```
#ifndef PACJENT_H
#define PACJENT_H
... (zawartość pliku)
#endif //PACJENT_H
```