

Podstawy programowania, Poniedziałek 15.05.2017

Zadanie 7

Piszemy program komputerowy dla Szpitalnego Oddziału Ratunkowego jednego ze szpitali. Na SOR często przyjmowani są ranni pacjenci, którym należy szybko przetoczyć odpowiednią ilość krwi. W naszym programie będziemy sprawdzać zgodność grup krwi pacjenta (biorcy) i potencjalnego dawcy.

Zakładamy, że grupy krwi są kodowane bitowo:

A+=101 (czyli 5), A-=100 (czyli 4), AB-=110 (czyli 6), O+=001 (czyli 1), B-=010 (czyli 2), 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: `int id`, `char *imię`, `char *nazwisko` `grupaKrw`, `uint8_t grupaKrw`. Umieść definicję struktury w pliku `Pacjent.h`.

Stwórz 2 funkcje powiązane ze strukturą Pacjent: `void UstawPacjent(Pacjent *pacjent)` oraz `void WypiszPacjent(Pacjent *pacjent)` ustawiające i wypisujące składniki struktury Pacjent. Następnie napisz funkcję `bool SprawdzZgodnosc(Pacjent *biorca, Pacjent *dawca)`, która zwraca nam zmienną typu `bool` (`true` lub `false`) w zależności od zgodności bądź nie grup krwi. Umieść deklaracje funkcji w pliku `Pacjent.h` a ich definicje w pliku `Pacjent.c`. Ustawiamy poleceniem `typedef` nazwę struktury na `Pacjent`. (3 p.)

Główna część zadania polega na stworzeniu dwóch pacjentów (biorcę i dawcę) z różnymi grupami krwi oraz sprawdzenie zgodności grup krwi poprzez użycie funkcji `SprawdzZgodnosc`. W wyniku operacji powinniśmy dostać wartość „true” lub „false”, w zależności od tego, czy porównywane grupy krwi są zgodne, czy nie, i wypisać właściwą informację na ekran. (2 p.)

Całość kompilujemy w linii poleceń używając pliku Makefile oraz polecenia make. (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
```

Uwaga 2!

Przykład piku Makefile

```
CC=gcc
```

```
CFLAGS=-Wall -pedantic -std=c99
```

```
LIBS=-lm
```

```
all: program
```

```
program: main.o struktura1.o struktura2.o
```

```
$(CC) -o main main.o struktura1.o struktura2.o $(CFLAGS) $(LIBS)
```

```
struktura1.o: struktura1.c struktura1.h struktura2.h
```

```
$(CC) -o struktura1.o -c struktura1.c $(CFLAGS)
```

```
struktura2.o: struktura2.c struktura2.h struktura1.h
```

```
$(CC) -o struktura2.o -c struktura2.c $(CFLAGS)
```

```
main.o: main.c struktura1.h struktura2.h
```

```
$(CC) -o main.o -c main.c $(CFLAGS)
```

```
clean:
```

```
rm -rf *.o main
```