Wstęp do programowania w języku C *(kolokwium 2014)*

1. (50p) Zakładamy, że w strukturze OSOBA zapisane są co najmniej: imię i nazwisko danej osoby, jej data urodzenia oraz dwa wskaźniki na struktury jej matki i ojca. Wskaźniki te mogą mieć w szczególności wartość NULL, jeśli w zbiorze danych odpowiednia osoba nie jest reprezentowana. Podać przykładową deklarację takie struktury. Następnie zdefiniować funkcję, która dla zadanych jako parametry wskaźnika na strukturę osoby oraz wskaźnika na pewne nazwisko sprawdzi, czy wśród przodków danej osoby występują osoby o podanym nazwisku i zwróci wskaźnik na jedną z nich. Jeśli takiej osoby nie ma w zbiorze danych, funkcja powinna zwrócić wartość NULL.

2. (50p) Silniową reprezentacją liczby naturalnej n, $1 \leq n < 36!$ nazywamy ciąg liczb $s_k, s_{k-1}, \ldots, s_1$, $1 \leq k \leq 35$, o następujących własnościach:

   (a) $n = s_k \cdot k! + s_{k-1} \cdot (k-1)! + \ldots, s_2 \cdot 2! + s_1 * 1!$;
   (b) $s_k > 0$ oraz $0 \leq s_i \leq i$ dla każdego $i \in \{1, \ldots, k\}$.

   Ze Wstępu do informatyki wiecie, że taka reprezentacja jest jednoznaczna. Napisać funkcję o nagłówku: `char *dodaj_silniowe(const char *liczba1, const char *liczba2)`, która dodaje dwie liczby podane w reprezentacji silniowej i tworzy wynik także w reprezentacji silniowej. W naszych reprezentacjach zakładamy, że wartości s_i ze zbioru $\{0, \ldots, 9\}$ zapisane są za pomocą odpowiedniego znaku '0', ..., '9', a wartości s_i ze zbioru $\{10, \ldots, 35\}$ za pomocą odpowiedniej dużej litery alfabetu łacińskiego: 'A', ..., 'Z', gdzie 'A' reprezentuje 10, 'B' - 11, itd. Pamięć na wynik funkcji powinna być przydzielana na stercie. Na przykład, wywołanie `dodaj_silniowe("54321","2001")` powinno zwrócić wskaźnik na `"102000"` . Funkcja powinna prawidłowo przetwarzać także bardzo duże liczby - takie, których wartość przekracza zakres typu `unsigned long long int`. Jeśli dane nie są poprawnie zapisanymi reprezentacjami silniowymi, funkcja powinna zwrócić wartość NULL.

   Wskazówka: Można zaimplementować "pisemną" metodę dodawania odpowiadających sobie cyfr silniowych z przeniesieniem.

3. (50p) Tablicę prostokątną $n \times m$ nazywamy *rzadką*, jeśli zawiera ona około 10% niezerowych wartości. Wiersz tablicy rzadkiej nazywamy *niepustym*, jeśli zawiera co najmniej jeden element niezerowy. Taką tablicę możemy pamiętać efektywnie jako listę niepustych wierszy, a w każdym niepustym wierszu - listę niezerowych elementów. Zdefiniować moduł RZADKA (tzn. podać zawartość plików `rzadka.h` i `rzadka.c`), w którym:

   (a) Za pomocą `typedef` zdefiniowany jest typ RZADKA reprezentujący tablicę rzadką o elementach typu `double`.
   (b) Zdefiniowana jest funkcja, która zwraca wartość z rzadkiej tablicy T występującą w wierszu i i kolumnie j.
   (c) zdefiniowana jest funkcja, która wpisuje wartość x do tablicy rzadkiej T w wierszu i i kolumnie j.
   (d) Zdefiniowana jest funkcja `RZADKA transponuj(const RZADKA T)`, która zwraca nową tablicę, w której w wierszu i i kolumnie j występuje taka sama wartość jak w tablicy `T` w wierszu j i kolumnie i, dla wszystkich par i i j. Tablica `T` powinna pozostać bez zmian.

9 stycznia 2014 *Marek Piotrów*