Lista zadań nr 1

Arytmetyka wskaźnikowa

Dynamiczna alokacja pamięci dla tablic jednowymiarowych i dwuwymiarowych

Zadanie 1

Pierwsze dwa argumenty funkcji `licz()` wskazują dwa elementy tej samej tablicy typu double. Uzupełnić jej definicję w taki sposób, by zwracała informację o liczbie elementów posiadających taką samą wartość jak trzeci argument i zawartych między elementami wskazywanymi przez pierwsze dwa argumenty włączając w to element wskazywany przez pierwszy argument oraz wyłączając element wskazywany przez drugi argument. Przetestuj funkcję w poniższym programie.

```
#include <stdio.h>
int licz(double *p1, double *p2, double x);
int main(void){
 double t[]={8.0, 2.0, 1.0, 6.0, 2.0, 7.0, 5.0, 2.0, 9.0};
 int n= licz(t+1, t+8, 2.0);
 printf("%d\n", n);
 return 0;
}
```

Zadanie 2

Napisz funkcję, która jako argumenty przyjmuje tablicę o elementach typu `double` oraz rozmiar tej tablicy. Funkcja powinna zwracać wskaźnik do elementu maksymalnego tej tablicy. W definicji funkcji nie możesz używać żadnych zmiennych typu `double` oraz korzystać z indeksowania. Przetestuj funkcję w prostym programie.

Zadanie 3

Napisz funkcję, która jako argumenty przyjmuje tablicę o elementach typu `int`, rozmiar tej tablicy oraz pewną liczbę całkowitą k. Funkcja powinna zwracać informację o liczbie elementów w tablicy, które są większej od k. W definicji funkcji możesz użyć tylko jednej zmiennej całkowitej i nie możesz używać indeksowania. Przetestuj funkcję w prostym programie.

Zadanie 4

Napisz funkcję `create_array()` przydzielającą pamięć dla tablicy o `k` elementach typu `double`. Liczba elementów tablicy jest parametrem tej funkcji, funkcja zwraca wskaźnik do odpowiedniego bloku pamięci zaalokowanego przy pomocy funkcji `malloc()`.

W programie korzystając z funkcji `create_array()` utwórz dwie dynamiczne tablice typu `double`: `t1` o rozmiarze `n` oraz `t2` o rozmiarze `m` (`n` i `m` pobierz w funkcji `scanf()`). W funkcji `complete_array()` wypełnij tablice `t1` i `t2` losowymi liczbami, a następnie

przekopiuj do tablicy o identyfikatorze t3 i rozmiarze n+m, również utworzonej dynamicznie. Wydrukuj tablice t1, t2 i t3 korzystając z funkcji print_array(). Pamiętaj o zwolnieniu przydzielonej pamięci z wykorzystaniem funkcji free().

Zadanie 5

Napisz funkcję reverse(), która jako argumenty przyjmuje dwa wskaźniki. Pierwszy powinien wskazywać na początkowy element pewnej tablicy liczb całkowitych (int), a drugi na ostatni element tej tablicy. Funkcja powinna zamieniać kolejno pierwszy element tablicy z ostatnim, drugi z przedostatnim itd., aż do elementów środkowych. Funkcja nie może korzystać z indeksowania tablicy, powinna zmieniać między sobą wartości wskazywane przez dwa wskaźniki.

Przetestuj funkcję reverse() w programie na 5 przykładowych tablicach tworzonych dynamicznie (skorzystaj z pętli). Rozmiary kolejnych tablic powinny być podawane przez użytkownika.

W celu utworzenia tablicy program powinien korzystać z funkcji create_array() zwracającej wskaźnik do odpowiedniego bloku pamięci zaalokowanego przy pomocy funkcji malloc(). Funkcja create_array() powinna również wypełniać utworzoną tablicę losowymi liczbami z zakresu od −50 do 50. W celu wyświetlenia tablicy program powinien korzystać z funkcji print_array(), która powinna drukować zawartość tablicy po 20 elementów w wierszu. Pamiętaj o zwalnianiu przydzielonej pamięci z wykorzystaniem z funkcji free().

Zadanie 6

Napisz funkcję bubble_sort(), która jako argumenty przyjmuje dwa wskaźniki. Pierwszy powinien wskazywać na początkowy element pewnej tablicy liczb całkowitych (int), a drugi na ostatni element tej tablicy. Funkcja powinna sortować tablicę niemalejąco wykorzystując algorytm sortowania bąbelkowego. Funkcja sortująca nie może korzystać z indeksowania. Wykorzystaj funkcję pomocniczą swap(), która dokonywać będzie zamiany wartości sąsiednich elementów tablicy, jako jej parametrów użyj również dwóch wskaźników.

Przetestuj funkcję bubble_sort() w programie na pięciu przykładowych tablicach tworzonych dynamicznie (skorzystaj z pętli). Rozmiary kolejnych tablic powinny być podawane przez użytkownika.

W celu utworzenia tablicy program powinien korzystać z funkcji create_array() zwracającej wskaźnik do odpowiedniego bloku pamięci zalokowanego przy pomocy funkcji malloc(). Funkcja create_array() powinna również wypełniać utworzoną tablicę losowymi liczbami z zakresu od −100 do 100. W celu wyświetlenia tablicy program powinien korzystać z funkcji print_array(), która powinna wyświetlać zawartość tablicy po 20 elementów w wierszu. Pamiętaj o zwalnianiu przydzielonej pamięci z wykorzystaniem z funkcji free().

Zadanie 7

Napisz program, w którym zdefiniowane zostaną następujące funkcje:

- funkcja `create_array()` - tworzy dynamicznie tablice dwuwymiarową liczb typu `int` o rozmiarach *n* na *m*, przekazywanych do funkcji przez argumenty jej wywołania, funkcja zwraca wskaźnik do zaalokowanego bloku pamięci;
- funkcja `complete_array()` - wypełnia tablice dwuwymiarową przekazaną jej jako argument liczbami losowymi z przedziału od 0 do 99, tablica jest pierwszym argumentem funkcji, drugim i trzecim argumentem wywołania funkcji powinny być wymiary tej tablicy;
- funkcja `print_array()` - wyświetla elementy tablicy dwuwymiarowej, o takich samych parametrach jak funkcja `complete_array();`
- funkcja `print_sum_rows()` - oblicza i wyświetla sumę elementów każdego wiersza tablicy dwuwymiarowej, o takich samych parametrach jak funkcja `complete_array();`
- funkcja `print_sum_columns()` - oblicza i wyświetla sumę elementów każdej kolumny tablicy dwuwymiarowej, o takich samych parametrach jak funkcja `complete_array().`

Przetestuj działanie wszystkich funkcji w programie dla wymiarów tablicy podanych przez użytkownika. Dynamicznego przydziału pamięci dla tablicy należy dokonać w oparciu o deklarację wskaźnika: `int **m;`

Zadanie 8

Napisz funkcję `swap_rows()`, która zamienia ze sobą dwa wybrane wiersze tablicy dwuwymiarowej - nie zamieniaj poszczególnych elementów w wierszach, wykorzystaj zamianę wskaźników, które przekażesz do funkcji jako argumenty. Wykorzystaj funkcję w programie z Zadania 7 zamieniając ze sobą pierwszy i ostatni wiersz tablicy dwuwymiarowej utworzonej dynamicznie.

Zadanie 9

Napisz funkcję `unique()` zwracającą informację o liczbie elementów różnych (niepowtarzających się) w tablicy jednowymiarowej liczb typu `int`, będącej parametrem funkcji. Wykorzystaj funkcję w programie z Zadania 7 sprawdzając, w którym wierszu tablicy dwuwymiarowej utworzonej dynamicznie znajduje się największa liczba elementów różnych.

Zadanie 10

Napisz funkcję zamieniającą w tablicy jednowymiarowej liczb typu `float` element maksymalny z elementem na pozycji j. Tablica oraz j są argumentami wywołania funkcji. Wykorzystaj tę funkcję w programie, który dla danej tablicy dwuwymiarowej kwadratowej o elementach typu `float`, ustawi element maksymalny w każdym wierszu na przekątnej głównej. Dla tablicy dwuwymiarowej należy dokonać dynamicznego przydziału pamięci (w

oparciu o wskaźnik `float (*m)[N];`). Program powinien również korzystać funkcji `complete_array()` wypełniającej tablicę dwuwymiarową liczbami losowymi oraz z funkcji `print_array()` wyświetlającej elementy tablicy dwuwymiarowej. Parametrami tych funkcji powinna być tablica oraz liczba jej wierszy.