

Lista 2 – wskaźniki

A) Zadania

1. Zaimplementuj poniższe funkcje. Do każdej dodaj prosty test poprawności:
 - a. // zwraca sumę size pierwszych elementów tablicy tab
`int suma(const int tab[], unsigned size);`
 - b. // to samo co powyżej, ale wynik przekazywany jest poprzez result
`void suma(const int tab[], unsigned size, int* result);`
 - c. // wypisuje size pierwszych elementów tablicy w strumieniu out, np. w `std::cout`
`void pisz(const int* tab, size_t size, std::ostream * out);`
 - d. // zapisuje w tablicy kolejne liczby całkowite, od 0 do size - 1
`void iota(int tab[], unsigned size);`
 - e. // odwraca kolejność size pierwszych elementów w tablicy tab
`void reverse(int* tab, size_t size);`
 - f. // oblicza iloczyn skalarny elementów 2 tablic o rozmiarze size
`int dot_product(const int* tab1, const int* tab2, size_t size);`
2. Napisz program, który przy pomocy funkcji `std::qsort` z biblioteki `<cstdlib>` porządkuje tablicę liczb typu `int` wg następującego kryterium: Najpierw rosnąco liczby parzyste, potem malejąco liczby nieparzyste. Na przykład ciąg
1, 4, 7, 2, 5, 9, 2, 8
po posortowaniu powinien przyjąć postać
2, 2, 4, 8, 9, 7, 5, 1.
Por.: <https://en.cppreference.com/w/cpp/algorithm/qsort> (tam jest przykład dość zaawansowany, z wyrażeniem lambda),
<http://www.cplusplus.com/reference/cstdlib/qsort/> etc.
Uwaga: `qsort` nie jest podstawową ani nawet zalecaną funkcją do sortowania w C++, ale jej interfejs

```
void qsort (void* base, size_t num, size_t size, int (*compar)(const void*,const void*));
```


zawiera masę wskaźników w przedziwnej konfiguracji i trzeba sobie z tym poradzić. W szczególności, `qsort` nie nadaje się do sortowania obiektów większości klas języka C++.

B) Problemy do wspólnego przedyskutowania na zajęciach

Można posłużyć się np. kompilatorem online https://www.onlinegdb.com/online_c++_compiler

1. Czy w prototypach funkcji z zadania A1 ma znaczenie, czy tablicę deklaruje się jako `int tab[]`, czy jako `int* tab`?
2. Zadeklaruj:
 - a. wskaźnik na zmienną typu `int`;
 - b. wskaźnik na wskaźnik z poprzedniego punktu
 - c. tablicę 10 zmiennych typu `double`
 - d. tablicę 10 wskaźników na zmienne typu `long long int`
 - e. wskaźnik na tablicę 10 zmiennych typu `double`
 - f. wskaźnik na tablicę 10 wskaźników na zmienne typu `char`
3. Niech `tab` i `p` będą zadeklarowane instrukcjami

```
int tab [10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
int* p = &tab[5];
```


Zakładając, że `int` zajmuje 4 bajty określ, jaką wartość ma wyrażenie:
 - a. `tab[1]`
 - b. `*tab`
 - c. `*(tab+1)`
 - d. `*p;`
 - e. `p[0]`
 - f. `p[3]`
 - g. `*(p+3)`
 - h. `*(p-1)`
 - i. `p[-1]`
 - j. `*&tab[0]`
 - k. `&tab[5] - &tab[0]`
 - l. `&tab[0] - tab`
 - m. `*(1+p)`
 - n. `p - tab`
 - o. `tab - p`

4. Opisz słowami, co deklarują poniższe instrukcje:
- `char (*tab)[10];`
 - `char *tab[10];`
 - `int tab[10][10];`
 - `double (*pf)(double x);`
 - `double (*pf)(double);`
 - `void qsort (void* base, size_t num, size_t size, int (*compar)(const void*, const void*));`
 - `void qsort (void*, size_t, size_t , int (*)(const void*, const void*));`
5. Jaką wartość mają poniższe wyrażenia (i dlaczego):
- `sizeof("Ala")`
 - `sizeof("\Ala\\")`
 - `sizeof("\n\t\n")`
 - `sizeof("Piec")`
 - `sizeof("Pięć")`

Uwaga: wyniki części z powyższych wyrażeń zależą od systemu (czy wiesz, dlaczego?). Obecnie w Linuksie standardem zapisu literału napisowego, np. "ę", jest UTF-8, w którym litera ę kodowana jest jako dwuznak 0xC499.

Literatura zalecana (do przejrzenia):

- <https://stackoverflow.com/questions/381542/with-arrays-why-is-it-the-case-that-a5-5a>
- <https://stackoverflow.com/questions/79923/what-and-where-are-the-stack-and-heap>
- <https://stackoverflow.com/questions/1143262/what-is-the-difference-between-const-int-const-int-const-and-int-const>
- <https://www.ioccc.org/>
- https://pl.wikipedia.org/wiki/Kodowanie_polskich_znak%C3%B3w