



Uniwersytet Jana Długosza w Częstochowie

Mykhailo Hulii

Informatyka, 2 rok,

Studia 1 stopnia

Grupa 2

A. Tablice

Zadanie 4

Napisać program obrazujący proste sortowanie tablicy jednowymiarowej o 15 elementach. (proste wybieranie lub proste zliczanie)

```
Task4!  
1 5 5 6 6 7 7 7 7 9
```

Zadanie 6

Napisz program wyświetlający środkowy element tablicy typu int (rozważ tablice z parzystą i nieparzystą ilością elementów) oraz jego indeks. Rozbuduj program o możliwość odwracania tablicy. W przypadku tablicy o parzystej ilości elementów środkowe elementy nie mogą zostać zamienione miejscami. Nie możesz używać tablicy pomocniczej a jedynie głównej oraz zmiennej pomocniczej typu int.

```
Task6!  
7, 1  
BEFORE  
8 0 0 7 7 1 3 7 3 1  
  
AFTER  
8 0 0 7 7 1 3 7 3 1
```

Zadanie 7

Napisz program generujący 10 000 liczb całkowitych z przedziału 0..1000. Program ma zliczyć ilość: zer, wartości 1000, ilość liczb parzystych i nieparzystych oraz podzielnych przez 3 a także sumę parzystych i nieparzystych. Program rozbudować o możliwość dokonywania serii np 20 losowań i podania wyników w postaci tabelarycznej.

```
Task7!  
count event: 847  
suma event: 847  
count not event: 153  
suma not event: 153  
count numbers % 3: 466
```

Zadanie 8

Napisać program, z zastosowaniem tablic, typu *Losowanie 6 z 49*. Program ma dokonywać trzech rodzajów losowań:

- 6 z 49 (mogą być powtórzenia)
- 6 z 49 (mogą być powtórzenia ale nie kolejno czyli w wylosowanym ciągu nie mogą się pojawić te same liczby obok siebie)
- 6 z 49 (bez powtórzeń)

```
6 out of 49 (with repetitions): 38 49 44 47 31 41
6 out of 49 (with repetitions but not adjacent): 41 14 20 22 3 11
6 out of 49 (without repetitions): 48 38 8 24 44 16
```

Zadanie 9

Napisać program będący emulatorem funkcji XOR dla wielu argumentów. Wykorzystać tablicę np 100 elementów.

```
XOR for 85 and for 85: 0
XOR for 85 and for 38: 1
XOR for 95 and for 85: 1
XOR for 85 and for 63: 1
XOR for 38 and for 38: 0
XOR for 39 and for 38: 1
XOR for 85 and for 85: 0
XOR for 39 and for 38: 1
XOR for 38 and for 63: 1
XOR for 38 and for 63: 1
XOR for 39 and for 48: 1
XOR for 63 and for 95: 1
XOR for -320846400 and for 83: 1
XOR for 39 and for -1588740032: 1
XOR for 85 and for 63: 1
XOR for 51 and for 83: 1
XOR for 51 and for 63: 1
XOR for -320846400 and for 36: 1
XOR for 36 and for 38: 1
XOR for 39 and for 2030045856: 1
XOR for -1588740032 and for -1588740032: 0
XOR for -1588740032 and for 63: 1
XOR for 36 and for 1: 1
XOR for 83 and for 23687: 1
XOR for 36 and for 2030045856: 1
XOR for -1588740032 and for 48: 1
XOR for 48 and for 30646: 1
XOR for 39 and for 30646: 1
XOR for 30646 and for 39: 1
XOR for 30646 and for 30646: 0
XOR for 0 and for 39: 1
XOR for -1588740032 and for 83: 1
XOR for 29 and for 38: 1
XOR for 51 and for 39: 1
XOR for 85 and for 0: 1
XOR for 39 and for -320856016: 1
XOR for 36 and for 0: 1
XOR for 23687 and for 36: 1
XOR for 0 and for 32764: 1
XOR for 83 and for -317387082: 1
XOR for 51 and for 30646: 1
```

Zadanie 12a

Napisz program, który przepisuje zawartość tablicy a do tablicy b, zamieniając kolumny na wiersze. Pierwsza tablica powinna być tak wypełniona danymi aby w pierwszej kolumnie były same jedynki, w drugiej dwójki itd. Tablica docelowa to: pierwszy wiersz same jedynki, drugi wiersz same dwójki itd.

```
Task12a!
First array:
982 816 379 217 497 774 715 479 960 778
183 33 778 561 716 183 490 442 646 203
817 433 307 444 507 771 378 647 281 717
659 263 885 38 480 382 164 547 213 477
325 397 862 104 310 578 639 152 21 285
355 838 71 14 634 578 786 13 225 67
730 237 682 967 627 162 702 144 61 915
621 386 664 483 490 974 61 129 478 82
767 833 273 190 200 259 768 986 272 346
405 354 583 87 322 210 601 24 706 662

Second array:
982 183 817 659 325 355 730 621 767 405
816 33 433 263 397 838 237 386 833 354
379 778 307 885 862 71 682 664 273 583
217 561 444 38 104 14 967 483 190 87
497 716 507 480 310 634 627 490 200 322
774 183 771 382 578 578 162 974 259 210
715 490 378 164 639 786 702 61 768 601
479 442 647 547 152 13 144 129 986 24
960 646 281 213 21 225 61 478 272 706
778 203 717 477 285 67 915 82 346 662
```

Zadanie 13

Napisz program wpisujący do macierzy 10x10 losowe wartości 0..99 ale tak aby na przekątnej (od lewej do prawej) pojawiały się kolejne wartości od 0..9. Program powinien wyświetlić całą macierz (na każdą liczbę zarezerwuj 3 pola, liczby dosunięte do prawej) oraz sumę elementów na przekątnej. Zoptymalizować program pod względem ilości instrukcji warunkowych.

Task13!

```
0 27 48 56 10 91 82 24 20 61
58 1 46 83 29 46 43 50 84 15
96 89 2 79 76 44 41 77 20 48
91 11 27 3 67 38 31 50 14 3
63 72 95 9 4 24 8 51 74 92
66 70 34 40 1 5 84 95 40 56
95 31 68 22 23 87 6 6 89 26
62 4 51 57 14 59 81 7 62 56
66 80 78 52 21 32 63 57 8 55
14 74 38 34 96 14 73 61 20 9
Task14!
```

Zadanie 14

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu. Zoptymalizować program pod względem ilości instrukcji warunkowych.

Task14!

```
39 82 67 42 39 33 53 73 7 9
81 26 96 59 78 17 43 93 8 22
0 40 96 39 74 45 53 7 6 25
63 45 8 82 88 99 6 93 72 75
9 5 53 5 17 5 74 60 29 0
83 30 41 31 4 67 76 26 67 34
51 82 80 3 65 20 63 33 13 87
60 74 2 66 31 10 50 5 70 79
58 1 61 99 37 82 18 13 8 38
0 12 72 80 23 89 52 86 22 17
```

Zadanie 14

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu. Zoptymalizować program pod względem ilości instrukcji warunkowych.

Task14!

```
39 82 67 42 39 33 53 73 7 9
81 26 96 59 78 17 43 93 8 22
0 40 96 39 74 45 53 7 6 25
63 45 8 82 88 99 6 93 72 75
9 5 53 5 17 5 74 60 29 0
83 30 41 31 4 67 76 26 67 34
51 82 80 3 65 20 63 33 13 87
60 74 2 66 31 10 50 5 70 79
58 1 61 99 37 82 18 13 8 38
0 12 72 80 23 89 52 86 22 17
```

Zadanie 15

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu. Zoptymalizować program pod względem ilości instrukcji warunkowych.

Before

```
6 5 2 9 1 5 1 1 1 1
2 9 9 4 0 6 8 0 1 7
0 1 1 3 1 4 2 5 3 7
5 9 2 9 0 3 4 1 6 7
4 0 8 3 6 8 1 5 1 3
4 1 6 5 4 0 9 9 5 4
6 0 5 8 9 5 3 4 8 1
1 3 1 2 8 8 0 0 5 3
5 9 7 1 4 3 1 5 2 9
0 8 9 5 8 1 3 3 7 3
```

After

```
0 25 -78 -78 -78 -78 -78 -78 -78 -78
1 81 -78 -78 -78 -78 -78 -78 -78 -78
2 1 -78 -78 -78 -78 -78 -78 -78 -78
3 81 -78 -78 -78 -78 -78 -78 -78 -78
4 0 -78 -78 -78 -78 -78 -78 -78 -78
5 1 -78 -78 -78 -78 -78 -78 -78 -78
6 0 -78 -78 -78 -78 -78 -78 -78 -78
7 9 -78 -78 -78 -78 -78 -78 -78 -78
8 81 -78 -78 -78 -78 -78 -78 -78 -78
9 64 -78 -78 -78 -78 -78 -78 -78 -78
```

Zadanie 18

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu.
Zoptymalizować program pod względem ilości instrukcji warunkowych.

Task18!

```
54 80 18 78 32 77 58 35 89 73
48 66 62 75 67 76 39 41 41 3
30 23 52 99 61 62 62 16 58 99
79 64 80 98 42 12 27 52 47 68
25 48 34 39 75 1 67 66 42 61
70 24 84 74 24 97 36 38 13 94
38 45 11 18 43 53 82 22 58 82
90 35 82 76 75 57 29 94 24 24
55 46 48 92 20 24 89 8 63 55
55 1 0 66 71 95 71 5 17 29
```

```
39 59 65 21 35 92 31 16 38 55
40 94 53 89 38 73 13 79 81 28
34 88 81 34 54 52 29 26 10 98
7 49 57 72 23 44 16 54 61 55
61 53 1 14 42 39 87 8 18 20
36 5 9 18 39 15 22 21 41 84
19 1 34 29 73 57 25 42 11 86
49 24 92 50 38 34 41 25 94 11
97 83 16 6 53 8 74 75 29 15
60 48 68 46 77 94 55 3 36 18
```

```
41 85 42 33 87 80 20 28 57 66
39 54 49 8 13 2 16 87 30 45
54 42 45 23 40 75 17 95 30 5
13 23 90 55 9 77 87 81 57 44
47 96 50 49 56 63 3 72 2 85
69 57 27 67 32 67 42 49 14 72
54 27 47 96 34 56 25 21 37 82
17 85 30 68 86 87 31 89 11 34
75 81 43 2 48 75 22 42 76 88
14 82 16 61 30 2 70 55 76 7
```

```
89 93 44 19 61 30 6 45 20 18
31 95 51 74 49 51 1 23 93 77
12 7 11 80 20 41 82 42 48 58
50 37 4 94 8 65 77 67 10 97
37 41 44 88 67 45 39 68 21 32
97 85 91 60 65 63 53 99 6 1
10 8 90 14 54 99 31 31 18 94
28 55 87 24 43 55 22 34 75 43
18 25 28 61 85 45 24 39 96 82
92 6 90 35 72 45 86 56 28 4
```


Zadanie 20

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu. Zoptymalizować program pod względem ilości instrukcji warunkowych.

```
Task20!  
Original string: hello world  
String after removing 'l': heo word
```

Zadanie 21

Zmodyfikuj program z zadania 13 tak aby przekątna była ustawiona od prawej do lewej od góry do dołu. Zoptymalizować program pod względem ilości instrukcji warunkowych.

```
Task21!  
Original destination: Hello  
Source: World!  
After appending: Hello World!
```

B. Struktury i unie

Zadanie 3a

Utworzyć program, który za pomocą funkcji tworzy:

- strukturę punkt (współrzędne x i y)
- strukturę kwadrat (współrzędne x1,y1, x2,y2)
- bada czy podany punkt (też jako struktura) zawiera się w kwadracie

Utworzone funkcje mają zwracać struktury, a funkcja badania punktu przyjmuje jako argumenty strukturę punktu i kwadratu. Utworzyć także funkcję normalizującą współrzędne kwadratu (funkcja tworzy nową strukturę, gdzie współrzędne niższe są przed wyższymi).

```
Task1!  
The point is not inside the square.  
After normalization, the coordinates of the square are (1, 5) and (1, 5)
```

Zadanie 4

Utworzyć strukturę *trojkat*, która przechowuje długości boków trójkąta. Napisać funkcję przyjmującą jako argument taką strukturę i zwracającą przez wartość obwód trójkąta. Całość użyć w programie, gdzie użytkownik wprowadzi dane długości poszczególnych boków.

```
Task2!
Enter the length of the 1 side of the triangle: 2
Enter the length of the 2 side of the triangle: 5
Enter the length of the 3 side of the triangle: 7
The perimeter of the triangle is: 14
```

Zadanie 4a

Napisać program prezentujący różnicę w przekazaniu do funkcji struktury przez wartość a przez wskaźnik. Struktura powinna zawierać zmienne: całkowitą, wskaźnik do całkowitej, wskaźnik do znaku, wskaźnik do ciągu znaków, string. Funkcja w swoim kodzie zmienia wszystkie dane dostarczonej struktury. Przed i po uruchomieniu funkcji wyświetlić zawartość struktury. Jak wygląda kwestia modyfikacji dostarczonych danych do funkcji? Jak taka modyfikacja wpływa na dane oryginalne? Które dane można w ten sposób zmienić czyli zwrócić z funkcji?

```
Task3!
Before calling functions:
Integer: 5
Integer pointed by intPointer: 10
Character pointed by charPointer: A
Character in charArrayPointer: e
String: Original

After calling modifyDataByValue:
Integer: 5
Integer pointed by intPointer: 20
Character pointed by charPointer: X
Character in charArrayPointer: Y
String: Original

After calling modifyDataByPointer:
Integer: 15
Integer pointed by intPointer: 60
Character pointed by charPointer: Z
Character in charArrayPointer: Y
String: Modified by pointer
```

Zadanie 13

Utworzyć strukturę z trzema polami bitowymi o wielkościach odpowiednio:

w_zakresie - jeden bit

maska - 32 bitow

adres_ip - 32 bitow

1- Wpisać do zmiennej w_zakresie wynik zapytania: czy 7 znajduje się w zakresach <-3..7) lub <7..21>

Jeśli wynik nie będzie poprawny zastanowić się czemu?

2- ustawić maskę adresu IP o długości 24 bity, podać numer podsieci i hosta dla adresu 192.168.0.130. Posłużyć się zapisem hexadecymalnym.

```
Is 7 within the ranges <-3..7) or <7..21>: Yes  
Subnet: c0a80000, Host: 82
```

Zadanie 17

Utworzyć nowy, złożony typ danych taki, aby do wprowadzonej zmiennej typu int można było się odnosić jak do czteroelementowej tablicy znaków (bajtów). Zakładamy, że typ int zajmuje 4 bajty.

```
Task5!  
Byte 1: 4  
Byte 2: 3  
Byte 3: 2  
Byte 4: 1
```

Zadanie 17a

inny sposób wykonania poprzedniego zadania:

Za pomocą unii z danymi typu unsigned short int i char[] sprawdzić co znajdziemy w zmiennej typu char po wpisaniu do zmiennej typu int wartości 65 (int leży w inii "nad" char)

```
Task6!  
Byte 1: 41  
Byte 2: 0  
Byte 3: 2  
Byte 4: 1
```

Zadanie 18

Napisać program obsługujący unię z czterema zmiennymi ułożonymi w następującej kolejności: int, float, int, char; wprowadzić dane dla pierwszej zmiennej i wyświetlić wartości wszystkich pól unii; powyższą czynność wykonać dla pozostałych pól.

```
Enter an integer for intValue1: 12
Values after setting intValue1:
intValue1: c
floatValue: 1.68156e-44
intValue2: c
charValue:
```

```
Enter a float for floatValue: 12.2
Values after setting floatValue:
intValue1: 41433333
floatValue: 12.2
intValue2: 41433333
charValue: 3
```

```
Enter an integer for intValue2: 24
Values after setting intValue2:
intValue1: 18
floatValue: 3.36312e-44
intValue2: 18
charValue:
```

```
Enter a character for charValue: a
Values after setting charValue:
intValue1: 61
floatValue: 1.35926e-43
intValue2: 61
charValue: a
```