 <p>Wrocław University of Science and Technology</p>	Sprawozdanie z programów z listy pierwszej		
	Student: <i>Jakub Król</i> 226269	Data Laboratoriów: <i>01.03.2017r. g. 16:30</i> Wykonano: <i>04.03.2017r.</i>	Prowadzący: <i>Dr inż. Krzysztof</i> <i>Halawa</i>
		Grupa laboratoryjna: <i>E02-18p</i>	Ocena:

Wstęp:

Programy polegały na przypomnieniu języka. Autor w dalszej części dokumentu będzie prezentował swoje programy **w języku C/C++**.

W kodzie nie ma komentarzy, jednak **wszelakie funkcje oraz zmienne są nazywane zgodnie z zastosowaniem w języku angielskim, tak aby kod był czytelny i przejrzysty**.

***Zgodnie z zaleceniami wykładowcy: Dr inż. Łukasz Jeleń, nie został wykorzystany STL** pomimo, że niektóre z programów można by napisać o wiele łatwiej z użyciem STL.

Wnioski nie są zbierane pod koniec dokumentu, są one wypisywane przy konkretnym zagadnieniu.

Program nr 1:

1. Napisz program implementujący tablicę dwuwymiarową alokowaną dynamicznie (rozmiar n wierszy i m kolumn), który będzie miał poniższe funkcje:

- (a) wypełniającą tablicę losowymi wartościami od 0 do x (x jest parametrem funkcji)
- (b) wyświetlającą zawartość tej tablicy,
- (c) znajdującą wartość maksymalną tablicy

Korzystając z przykładu na stronie dr. J. Mierzwy proszę stworzyć do powyższego zadania odpowiednie menu.

Opis:

Polecenie nie jest zbyt szczegółowe. **Brakuje m.in. rodzaju typu wejścia oraz formatu wyjścia.** W podpunktach jest mowa o wartościach, dlatego też do wyboru jest typ całkowity lub zmiennoprzecinkowy. **Autor postanowił oprzeć program o liczby całkowite, z powodu braku określonego wejścia została użyta zmienna typu long int.**

Z powodu, że program udostępnia interfejs użytkownikowi, **zostały wprowadzone zabezpieczenia przed nieprawidłowymi danymi lub akcjami.**

Kod źródłowy:

Z powodu braku czytelności, kod programu został zahostowany w serwisie pastebin.com

<http://pastebin.com/c7mTaYS0>

Testy programu:

```
- - - - -  
1. Ustal rozmiar tablicy  
2. Ustal maksymalna wartosc x  
3. Wypelnij tablice wartosciami od 0 do x  
4. Wyszwietl tablice  
5. Usun tablice  
0. Koniec (usuwa dynamiczna tablice)  
- - - - -
```

Wybor (0-5): 1

Podaj liczbe wierszy: 5
Podaj liczbe kolumn: 10

Wybor (0-5): 1

Podaj liczbe wierszy: 5
Podaj liczbe kolumn: -5
Podano nieprawidlowe wymiary tablicy

Wybor (0-5): 1

Podaj liczbe wierszy: 5
Podaj liczbe kolumn: 0
Podano nieprawidlowe wymiary tablicy

Wybor (0-5): a

Brak takiej opcji w menu

W poleceniu zostało wyraźnie zaznaczone, że ma być to tablica dwuwymiarowa, także wszelkie wymiary ≤ 0 są niedopuszczalne.

Program został również zabezpieczony przed wprowadzeniem nieprawidłowej opcji menu.

```
Wybor (0-5): 2  
Podaj wartosc x: 5
```

```
Wybor (0-5): 2  
Podaj wartosc x: -5  
Podano nieprawdilowa wartosc x, podaj wartosc >0
```

Program ma za zadanie wypełniać tablice wartościami losowymi z zakresu od 0 do x. Dlatego też wypełnianie losowe wartościami z zakresu 0-0 nie miało by sensu. Dodatkowo, zostało wprowadzone zabezpieczenie przed wprowadzeniem liczby ujemnej.

```
Wybor (0-5): 3  
Nie podano rozmiaru tablicy badz wartosci x
```

```
Wybor (0-5): 3
```

Jeżeli nie został wprowadzony rozmiar tablicy, bądź nie została wprowadzona wartość x, opcja 3 nie będzie dostępna. Jeżeli nie ma zastrzeżeń, program nie wyświetla żadnego komunikatu.

```

- - - - -
1. Ustal rozmiar tablicy
2. Ustal maksymalna wartosc x
3. Wypelnij tablice wartosciami od 0 do x
4. Wyszwietl tablice
5. Usun tablice
0. Koniec (usuwa dynamiczna tablice)
- - - - -

Wybor (0-5): 4

Nie wypelniono tablicy

```

```

- - - - -
Wybor (0-5): 5

Tablica nie istnieje
- - - - -

```

Zabezpieczenie w przypadku opcji 4 oraz 5 ma się analogicznie co w przypadku opcji numer 3.

```

- - - - -
Wybor (0-5): 4

      1      2      4      0      4      4      3      3      2      4
      0      0      1      2      1      1      0      2      2      1
      1      4      2      3      2      2      1      1      3      0
      2      1      1      3      4      2      2      4      0      4
      3      1      2      3      3      4      1      1      3      3

```

Wprowadzono wymiary 5x10 oraz wartosc maksymalna 5, nastepnie wypelniono tablice oraz wybrano opcje numer 4 aby ja wyswietlic. Tablica jest formatowana pod liczby max 5 cyfrowe, jednak mozna bez problemu zmienic to w kodzie programu.

Jest to przykladowy sposob realizacji, gdyz brakowalo opisu w tresci zadania. Wyświetlanie mozna by rowniez zrealizowac w sposob zalezny od dlugosci podanej wartosci „x”, jednak wymagałoby to dodatkowych funkcji.

Program nr 2:

2. Dla tablicy jednowymiarowej należy stworzyć funkcje (mogą stanowić rozszerzenie programu z zadania 1):

- (a) zapisująca tablicę jednowymiarową do pliku tekstowego
- (b) wczytującą tablicę jednowymiarową z pliku tekstowego
- (c) zapisującą tablicę jednowymiarową do pliku binarnego
- (d) wczytującą tablicę jednowymiarową z pliku binarnego

Proszę stworzyć odpowiednie menu dla tych funkcji.

Opis:

Podobnie jak w przypadku 1 zadania **brakuje tutaj opisu wejścia i wyjścia**. Początkowym pomysłem autora było oparcie programu o typ string, aby można było wczytywać każdy rodzaj danych, jednak **tablica typu string jest już tak naprawdę tablicą dwuwymiarową**. Aby zastosować się do polecenia (tablica jednowymiarowa) **wykorzystano typ long int** (niestety wyklucza to wczytywanie danych typu „char”).

Najłatwiej byłoby oprzeć program o STL vector, jednak na wykładzie zostało to zabronione, dlatego cofnięto się do języka C wykorzystując realloc();

Program udostępnia menu, dlatego też **zabezpieczono przed wprowadzeniem nieprawidłowych danych/poleceń**.

Program nie otworzy nie istniejącego pliku, jednak przy zapisie, jeżeli plik nie istnieje to go utworzy, jeżeli plik istnieje dopisze on dane na końcu pliku.

Kod źródłowy:

Z powodu braku czytelności, kod programu został zahostowany w serwisie pastebin.com

<http://pastebin.com/gQg1Dfe3>

Testy programu:

```
- - - - -
1. Wczytaj tablice z pliku tekstowego
2. Wczytaj tablice z pliku binarnego
3. Zapisz tablice do pliku tekstowego
4. Zapisz tablice do pliku binarnego
0. Wyjdź z programu
- - - - -

Wybor (0-4): a
Brak takiej opcji w menu
```

Program jest zabezpieczony przed podaniem nieprawidłowej opcji w menu

```
Wybor (0-4): 1
Podaj nazwe pliku: nie_istniejacy.txt
Cos poszlo nie tak, sprawdz czy plik istnieje lub nie jest uzywany przez inny pr
ogram
```

```
Wybor (0-4): 2
Podaj nazwe pliku: nie_istniejacy.bin
Cos poszlo nie tak, sprawdz czy plik istnieje lub nie jest uzywany przez inny pr
ogram
```

```
- - - - -
Wybor (0-4): 1
Podaj nazwe pliku: L1_Z2_pliktxt.txt
- - - - -
```

Program nie wczyta pliku który nie istnieje, bądź jest używany przez inny program.

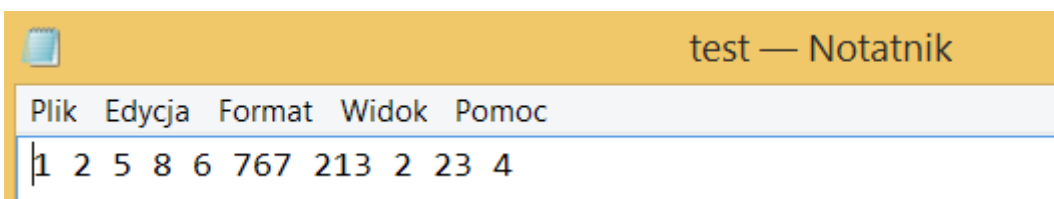
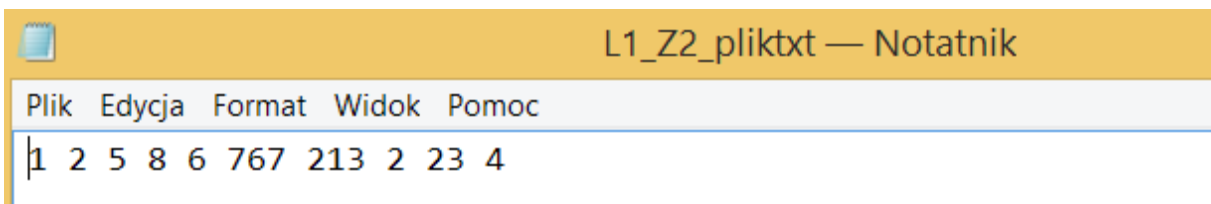
W przypadku prawidłowego wczytania, program nie zwraca żadnej informacji.

```

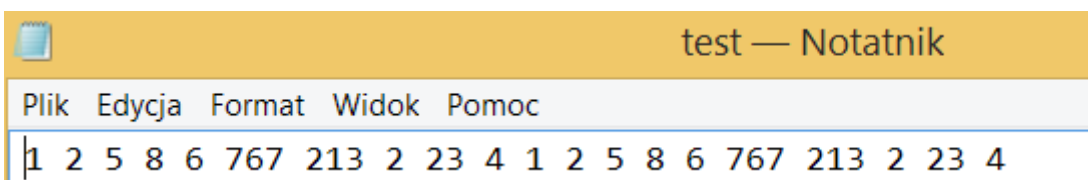
- - - - -
Wybor (0-4): 1
Podaj nazwe pliku: L1_Z2_pliktxt.txt
- - - - -
1. Wczytaj tablice z pliku tekstowego
2. Wczytaj tablice z pliku binarnego
3. Zapisz tablice do pliku tekstowego
4. Zapisz tablice do pliku binarnego
0. Wyjdź z programu
- - - - -

Wybor (0-4): 3
Podaj nazwe pliku: test.txt
- - - - -

```



Wczytanie pliku tekstowego i zapis do pliku tekstowego działa poprawnie.



Jeżeli zostanie wywołany plik już istniejący, który nie jest pusty, program dopisze swoje wyjście na koniec pliku.

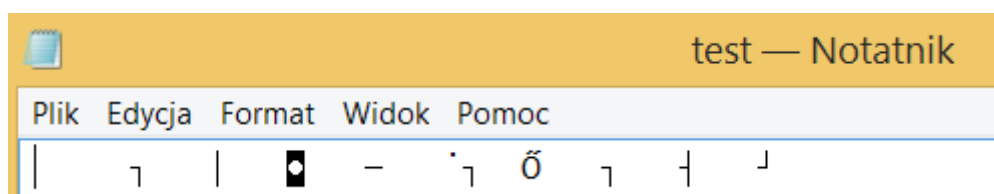
Wybor (0-4): 1

Podaj nazwe pliku: L1_Z2_pliktxt.txt

```
- - - - -  
1. Wczytaj tablice z pliku tekstowego  
2. Wczytaj tablice z pliku binarnego  
3. Zapisz tablice do pliku tekstowego  
4. Zapisz tablice do pliku binarnego  
0. Wyjdź z programu  
- - - - -
```

Wybor (0-4): 4

Podaj nazwe pliku: test.bin



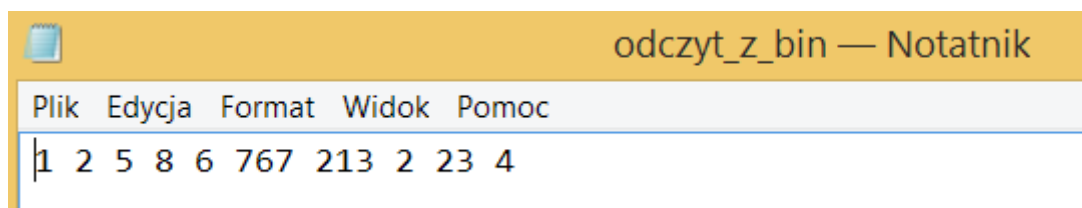
Wybor (0-4): 2

Podaj nazwe pliku: test.bin

```
- - - - -  
1. Wczytaj tablice z pliku tekstowego  
2. Wczytaj tablice z pliku binarnego  
3. Zapisz tablice do pliku tekstowego  
4. Zapisz tablice do pliku binarnego  
0. Wyjdź z programu  
- - - - -
```

Wybor (0-4): 3

Podaj nazwe pliku: odczyt_z_bin.txt



Zapis do pliku binarnego jak i zarówno odczyt z pliku binarnego działa poprawnie.

Program nr 3:

3. Należy stworzyć funkcje rekurencyjną wyznaczającą:

(a) x^p - prototyp funkcji: `int Potega(int x, int p);`

(b) $x!$ - prototyp funkcji: `int Silnia(int x);`

Opis:

Są to tylko funkcje rekurencyjne bez programu, w tym dokumencie załączone są tylko i wyłącznie funkcje. Testy zostały przeprowadzone poprzez dopisanie programu testującego.

W przypadku funkcji a) został zmieniony prototyp na `float potega (int x, int p)`. Z powodu, że `int p` pozwala na potęgowanie z liczbami ujemnymi, które są wynikowo liczbami zmiennoprzecinkowymi.

Kod źródłowy funkcji rekurencyjnych:

```
1. long int silnia(int x)
2. {
3.     if (x<0) cerr << "Nie mozna wyliczyc silnii z liczby ujemnej" << endl;
4.     if (x<2) return 1;
5.
6.     return x*silnia(x-1);
7. }
8.
9. float potega(int x, int p) // algorytm szybkiego potegowania!
10. {
11.     if (p==0) return 1;
12.
13.     if (p%2!=0){
14.         if (p>0) return x*potega(x,p-1);
15.         if (p<0) return (1.0/x)*potega(x,p+1);
16.     } else {
17.         float tmp = potega(x,p/2);
18.         return tmp*tmp;
19.     }
20. }
```

Testy funkcji rekurencyjnych:

```
Podaj x: 5  
Podaj p: 5  
  
n! = 120  
x^p = 3125
```

```
Podaj x: 5  
Podaj p: -5  
  
n! = 120  
x^p = 0.00032
```

```
Podaj x: -5  
Podaj p: 5  
x^p = -3125
```

Funkcja silnia działa prawidłowo.

Potęgowanie liczb dodatnich oraz ujemnych działa prawidłowo.

Dodatkowo, potęgowanie przy ujemnych potęgach działa również prawidłowo.

Algorytm szybkiego potęgowania zmniejsza wymagany czas przy dużych liczbach.

Program nr 4:

4. Palindromem nazywamy takie słowo, które czytane z obu stron jest takie samo. Należy napisać funkcję rekurencyjną do przetestowania łańcucha znakowego w celu sprawdzenia, czy jest on palindromem. Prototyp takiej funkcji to `bool jestPal(String testStr);`. Np.: Słowa „kajak”, czy „1234321” są palindromami i dla nich funkcja powinna zwrócić `true`, ale słowo „PAMSI” już nim nie jest i dla niego funkcja powinna zwrócić `false`.

Proszę zapisać ten program, ponieważ na kolejnych zajęciach będziemy dopisywali do niego dodatkową funkcjonalność.

Opis:

Program wypisuje czy dane słowo jest palindromem czy nie. Wykorzystuje do tego funkcję rekurencyjną. Został użyty prototyp tak jak w poleceniu. **W poleceniu nie ma nic na temat dużych oraz małych liter, które mają różne oznaczenie ASCII**, dlatego też autor programu postanowił rozszerzyć funkcjonalność programu. Zarówno słowo „kajak” jak i „Kajak” będą uznawane jako palindrom.

Kod źródłowy:

```
1. #include <iostream>
2. #include <cctype>
3.
4. using namespace std;
5.
6. bool check_palindrom(string word)
7. {
8.     bool status;
9.     int len=word.length();
10.    if (len<=1) return true;
11.
12.
13.    if(word[0]==word[len-1]){
14.        status=true;
15.        word.erase(len-1,len-1);
16.        word.erase(0,1);
17.    }
18.    else status=false;
19.
20.    return status && check_palindrom(word);
21. }
```

```

22.
23. int main()
24. {
25.     string word;
26.     cin >> word;
27.
28.     string temp_word=word;
29.     for (unsigned int i = 0; i < temp_word.length(); i++)
        temp_word[i]=tolower(temp_word[i]);
30.
31.     check_palindrom(temp_word) ? cout << "Słowo: " << word << " jest palindromem" : cout <<
        "Słowo: " << word << " nie jest palindromem";
32.
33.     return 0;
34. }

```

Testy programu:

```

12321
Słowo: 12321 jest palindromem
-----

```

```

kajak
Słowo: kajak jest palindromem
-----

```

```

Kajak
Słowo: Kajak jest palindromem
-----

```

```

PAMSI
Słowo: PAMSI nie jest palindromem
-----

```

Program bez problemu rozpoznaje co jest, a co nie jest palindromem. Dodatkowo dobrze radzi sobie z przypadkiem, gdy są używane w jednym słowie duże i małe litery.