Katedra
Informatyki i Automatyki
Politechniki Rzeszowskiej

# Informatyka

## 4.Wstęp do programowania w C/C++

Opracował: Maciej Penar

# Spis treści

# 1. Wprowadzenie do C

Napisać w **C** aplikacje konsolowe:

1. Zapytaj użytkownika o imię, nazwisko i wiek. Wyświetlić „Witaj [imię] [nazwisko], masz [wiek] lat”.
   - Zwróć uwagę na walidację wejścia: wiek musi mieć sens, imię/nazwisko nie mogą być puste.
   - *Dla chętnych, **można skorzystać z C++**: rozszerzyć walidację w zadaniu 1 w oparciu o wyrażenia regularne tak żeby imię i nazwisko zaczynało się od dużej litery i składało się tylko z liter.*

| Podpowiedź do 1b |
| --- |
| Wyrażenia regularne, ściągawka: https://regexone.com/<br>Dokumentacja link: http://en.cppreference.com/w/cpp/regex)<br>W sumie trzy linijki:<br>Pierwsza: #include <regex><br>Druga: std::basic_regex<char> reg(CAŁA ZAGADKA, std::regex::icase);<br>Trzecia: `if (std::regex_match<char>(text, reg))` |

2. Brytyjczycy są dziwni i podają wagę w kamieniach. 1 kamień = 6.35029318 kilograma. Napisz program który pyta użytkownika o jednostkę w jakiej poda wagę oraz wartość wagi. Wypisz przeliczenie na kilogramy/kamienie. Użyj stałych symbolicznych. Napisz dwie wersje programu (lub jedną sprytną w której możliwa jest łatwa podmiana typu) – wykorzystującą double i float. Czy są różnice.
3. Napisz funkcję wykonującą potęgowanie liczby 2 (wykładnik to parametr):
   - Przez pętle
   - Przez przesunięcie bitowe
   - *Dla chętnych: dodać implementację przez pseudo-przesłonięcie funkcji pow() z <math.h>. Wykonać test która z trzech metod jest szybsza.*

| Podpowiedź do 3c |
| --- |
| Dołączyć <time.h>.<br>Poniższy kod wylicza różnicę czasu:<br>`time_t start,end;`<br>`double dif;`<br><br>`time (&start);`<br>`// Do some calculation.`<br>`time (&end);`<br>`dif = difftime (end,start);`<br><br>Link: https://stackoverflow.com/questions/3557221/how-do-i-measure-time-in-c |

4. Napisz funkcję która oblicza silnie: n!
   - Napisz wersję iteracyjną (wykorzystującą pętle)
   - Napisz wersję rekurencyjną (funkcja wywołująca samą siebie). Spróbuj wykorzystać *„operator trójargumentowy”*
   - Wyrób opinię na temat obu wersji (długość kodu / czytelność kodu / wydajność)
   - Dla jakiego x następują anomalie na typach int, long, long long

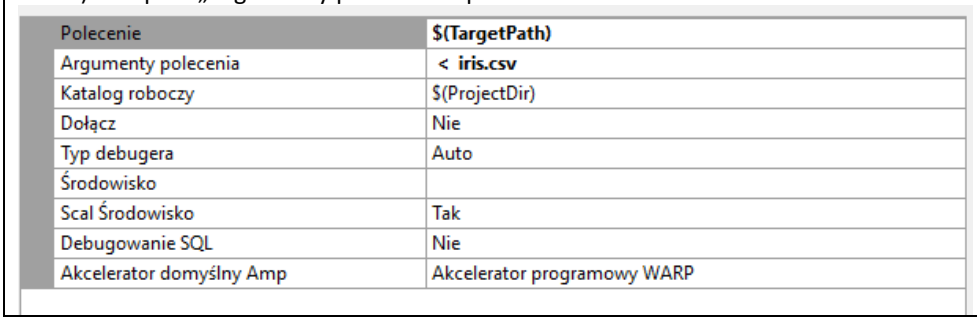
| Podpowiedź do 4/4b |
| --- |
| Operator trójargumentowy ma następującą składnie:<br>[warunek] ? [wartość gdy prawda] : [wartość gdy fałsz]<br>Np. zapis **x > 0 ? x : -x** oblicza wartość bezwzględną liczby x<br><br>Wzór na silnie:<br>$factorial(x) = x! = x * (x-1) * (x-2) \ldots * 2 * 1 = \prod_{i=1}^{x} i$ |

5. Napisać strukturę *student* która przechowuje następujące dane: identyfikator, imię, nazwisko, wiek, rok studiów, semestr. Dobrać typy danych oraz uzasadnić ich wybór.
6. Kontynuacja (5): Okazuje się że student może być identyfikowany albo za pomocą:
    - Liczby nieujemnej całkowitej
    - 10 znaków

Zaproponować unię (link: https://www.tutorialspoint.com/cprogramming/c_unions.htm) reprezentującą identyfikator. Zainicjalizować unię znakami, sprawdzić:

a) Adres poszczególnych pól
b) Szerokość unii
c) Wartość znaków
d) Wartość liczbową

7. Dany jest plik **iris.csv** (dość popularny, jest tu np. https://github.com/mpenarprz/InformatykaA1/blob/master/dane/iris.csv). Pobrać. Przekierować plik jako dane wejściowe do programu. Zaproponować **strukturę** do przechowywania danych. Napisać program który wyliczy średnią wartość pól „sepal_length", „sepal_width", „petal_width", „petal_length" w każdej z grup „species". Wypisać te wartości. Zastanowić się czy trzeba trzymać wszystkie obiekty w pamięci. Wyrzucić nagłówek pliku .csv (można ręcznie).

| Podpowiedź do 7 |
|---|
| W Visual Studio by przekierować plik jako wejście wczytywane z scanf() / cin należy:<br>a) Skopiować plik do folderu w którym jest projekt<br>b) Wejść we właściwości projektu (PPM->Właściwości)<br>c) Zakładka Debugowanie<br>d) W polu „Argumenty polecenia wpisać **< iris.csv**<br>  |

| Polecenie | **$(TargetPath)** |
|---|---|
| Argumenty polecenia | **< iris.csv** |
| Katalog roboczy | $(ProjectDir) |
| Dołącz | Nie |
| Typ debugera | Auto |
| Środowisko | |
| Scal Środowisko | Tak |
| Debugowanie SQL | Nie |
| Akcelerator domyślny Amp | Akcelerator programowy WARP |

8. Kontynuacja 7: jeśli poprzednie zadanie wykonano z wyliczaniem „w locie" – przerobić program tak by buforował rekordy.

# 2. Obok C

| Podpowiedź do kolejnych zadań (1,2) |
| --- |
| W Visual Studio można zmusić pre-processor by zapisał wynik do pliku, by to wykonać należy:<br>1. Wejść w ustawienia projektu (alt + enter)<br>2. Właściwości konfiguracji -> C/C++ -> Preprocessor<br>3. Zaznaczyć **Przetwarzaj Wstępnie do Pliku TAK** |

Odpowiedz na pytania:

1. Co robi dyrektywa #include
2. Co robi dyrektywa #define:

```
#define MAX(a,b) (a>b ? a:b)
void f(){
  if (MAX(-1, 1) > 0) {…}
}
```

3. Jaki będzie efekt działania kodu:

```
int i = 0;
printf("%d", i);
printf("%d", i++);
printf("%d", ++i);
sizeof(i++);
printf("%d", i);
```

4. Przedstaw program któremu kończy się pamięć:
    a. Na stercie (Heap)
    b. Na stosie (Stack)
5. Oceń kod:

```
#include <iostream>
template
<typename T>
T pomnoz(T a, T b) {
	auto output = 0;
	auto lim = b >= 0 ? b : -b;
	for (int i = 0; i < lim; ++i)
		output += a;
	return b < 0 ? -output : output;
}

int main()
{
	int zmienna;
	for (int i = -10; i <= 10; ++i) {
		for (int j = -10; j <= 10; ++j) {
			zmienna = pomnoz<int>(i, j);
			printf("<%d,%d>: %d\n", i, j, zmienna);
		}
		std::cout<< std::endl;
	}
}
```

6. Na czym polega przepełnienie liczb całkowitych (Integer Overflow). Napisać program który ilustruje problem. Wykorzystać liczby całkowite ze znakiem oraz bez znaku.