Projekty zaliczeniowe
Podstawy Programowania 2012/2013

0. Zasady ogólne

W skład projektu wchodzą następujące elementy:

- dokładny opis rozwiązywanego problemu
- opis słowny rozwiązania problemu wraz z pseudokodami
- szacunkowy opis efektywności (ilość operacji wykonywanych przez aplikację w zależności od danych wejściowych)
- instrukcja obsługi
- kod źródłowy aplikacji
- przykładowe dane wejściowe (jeśli dotyczy)

Wszystkie materiały poza źródłami i danymi wejściowymi winny zostać dostarczone w jednym pliku PDF.

1. Labirynt

Celem aplikacji jest wczytanie (z pliku bądź ze standardowego wejścia) dwuwymiarowego obrazu labiryntu o określonych rozmiarach. Każde pole może być oznaczone przez jedną z wartości:
- '#' - ściana, nie można na nią wejść
- '.' - korytarz, można po nim chodzić
- '$' - punkt startowy
- '@' - punkt docelowy.

Aplikacja na wyjściu musi wypisać ciąg kroków, nieoddzielonych żadnymi znakami (jak spacje, nowe linie), który wystarczy do przejścia labiryntu od początku do końca. Droga nie musi być optymalna, dopuszczane są nawroty.

Wejście:
- dwie liczby oznaczające wysokość i szerokość labiryntu
- obraz labiryntu

Np:

```
5 6
#$.###
##...#
#..###
##..##
###.@#
```

Wyjście:

ciąg kroków, gdzie:
- L - krok w lewo
- P - krok w prawo
- D - krok w dół
- G - krok w górę.
- S - start
- K - koniec

Np:
SPDPPLLDDPDPK

Wersja uproszczona:

Algorytm może zakończyć się w ślepym zaułku i już nie poszukiwać dalszego wyjścia.
Dodajemy wtedy nowy stan
- B - blokada

Przykład dla labiryntu powyżej:
SPDDLB

2. Gra planszowa/strategiczna

Zrealizowana w konsoli lub prostym systemie graficznym (np. PDCurses lub OpenGL + GLUT). Zasady gry i metoda wykonania do dyskusji. Szczegóły i poziom trudności projektu do ustalenia.

Wersja uproszczona:

- brak systemu graficznego, tylko czyszczona za każdym ruchem konsola.

3. Wyszukiwanie najkrótszej ścieżki.

Dana jest tablica dwuwymiarowa, zawierająca liczby dodatnie.
Należy znaleźć ścieżkę od pierwszej kolumny do ostatniej, taką, by suma pól, przez które wiedzie, była jak najmniejsza. Można poruszać się na boki i po skosie. Co więcej, z pierwszego wiersza można przejść do ostatniego. Na wejściu podane są wymiary tablicy i sama tablica, na wyjściu pojawić ma się suma odwiedzonych pól, nie trzeba wypisywać poszczególnych kroków. Dane testowe można uzyskać od prowadzącego zajęcia.

Przykład:

Wejście:

3 4

6 2 5 4
3 9 3 2
4 4 1 3

Wyjście:

Suma pól - 8. (Ścieżka wiedzie przez pola 3, 2, 1 i 2.)

Wersja uproszczona:

Jako pierwsza liczba na wejściu podawana jest liczba kroków na przód, którą program musi analizować. Z tych kroków za każdym razem wybiera optymalną w danej chwili drogę, nie musi być ona najlepsza w kontekście całej tablicy. Przy remisach wybór drogi dowolny. Między pierwszym rzędem a ostatnim NIE MA połączenia.

Przykład:

2 //ilość kroków
5 5

3 3 1 1 1
6 6 6 6 6
6 6 6 6 6
1 1 6 6 6
6 6 6 6 6

Wyjście:
Suma pól - 20

Wyjaśnienie:
Program bada za każdym razem dwa najbliższe kroki. Dzięki temu wybiera drogę przez rząd czwarty, zamiast przez pierwszy.

Wersja bardzo uproszczona:

Program bada tylko koszt najbliższego pola, przy czym za każdym krokiem musi poruszać się w prawą stronę.

4. Formatowanie pliku

Dla podanego pliku wejściowego zastosuj formatowanie zgodnie z poniższymi regułami. Sformatowany tekst wypisz na ekran.

Wersja uproszczona:

- po przecinku zawsze jest spacja ("tekst,tekst" => "tekst, tekst")
- dookoła nawiasów okrągłych są spacje ("asdasdas(dasdasdas)asdasd"=>"asdasdas (dasdasdas) asdasd")
- wewnątrz nawiasów kwadratowych są spacje ("asdasd[qwe]asdasd" => "asdasd[qwe]asdasd")

- operatory matematyczne (+, -, *, /, %, =), logiczne (<, >, >=, <=, ==, &&, ||) i bitowe (|, &, <<, >>) otoczone są spacjami
- nawiasy klamrowe są jedynymi niebiałymi znakami w linii (otoczone są znakami nowej linii)
- znak średnika kończy linię
- wszystkie tabulatory na początku linii zamienione są na 4 spacje
- nie ma podwójnych spacji poza początkiem linii ("tekst tekst" => "tekst tekst")

Wersja zaawansowana:

- reguły wersji uproszczonej
- średnik NIE kończy linii, jeżeli jest otoczony nawiasami okrągłymi (jak np. w pętli for)
- wcięcia:
  - jeden poziom wcięcia to cztery spacje
  - otwarcie nawiasu klamrowego zwiększa poziom wcięć o jeden w następnej linii
  - zamknięcie nawiasu klamrowego zmniejsza poziom wcięcia w linii z nawiasem

Przykład:

```
{test{asd}asd}
=>
{
    test
    {
        asd
    }
    asd
}
```

5. Wyszukiwanie tekstu

Program przeszukuje wszystkie pliki *.txt z bieżącego katalogu w poszukiwaniu wzorca podanego z klawiatury. Wzorzec może zawierać * jako oznaczenie ciągu dowolnych znaków (również ciągu pustego) lub ? jako oznaczenie dokładnie jednego dowolnego znaku. Nie dotyczy to znaków końca linii.

Program ma wypisać nazwę pliku, numer linii, w której znaleziono zadany tekst, oraz faktycznie znaleziony tekst.

Wersja uproszczona:

przeszukiwanie tylko jednego pliku, o nazwie podanej z klawiatury.

6. Wyszukiwanie permutacji - projekt prosty

W pliku wejściowym, w pierwszej linii, znajduje się ciąg liczb dodatnich. Należy w dalszych liniach znaleźć wszystkie wariacje (bez powtórzeń) tego ciągu i zapisać je do pliku wyjściowego.

7. Zliczanie wyrazów - projekt prosty

Program ma za zadanie zliczyć liczbę wystąpień wyrazów w wejściowym pliku tekstowym oraz wypisać je na ekran w kolejności od najczęstszych do najrzadszych.

8. Quiz - projekt prosty

Program ma za zadanie wczytać pytania z pliku tekstowego (format dowolny), wyświetlić je w losowej kolejności (wraz z losową kolejnością odpowiedzi) oraz zebrać odpowiedzi użytkownika. Po zakończonym quizie, program wyświetla liczbę zdobytych punktów (również procentowo) oraz listę pytań, na które odpowiedziano źle (wraz z zaznaczeniem odpowiedzi poprawnej i tej, której udzielił użytkownik). Format graficzny dowolny. Pytania nie mogą się powtarzać.

9. Trener pisania na klawiaturze

Program wczytuje bazę zdań z pliku, każde zdanie w osobnej linii. Następnie wyświetla użytkownikowi jedno zdanie i każe je przepisać. Użytkownik wpisuje tekst, jednak nie widzi wpisanego tekstu na ekranie. Po naciśnięciu "enter" program wyświetla wpisane przez użytkownika zdanie, wyraźnie pokazuje błędy oraz pokazuje procentową ilość błędów (za dobre uznajemy właściwe litery na właściwych pozycjach). Do tego program pokazuje czas, jaki upłynął od naciśnięcia pierwszego klawisza zdania do ostatniego.

10. Szyfrowanie plików wybraną metodą podstawieniową

Do wyboru szyfry ADFGVX, Playfair, Vigenere, Four-square lub inne – do ustalenia z prowadzącym.

11. Kompresja pliku tekstowego metodą RLE – projekt prosty

Program wczytuje nazwę pliku wejściowego i wyjściowego oraz tryb działania (kompresja albo dekompresja). W zależności od trybu, w pliku wyjściowym znaleźć ma się przetworzony plik wejściowy. Plik wejściowy składa się tylko z liter i znaków nienumerycznych.