

Podstawy programowania 2008/2009. Kolokwium 2, Grupa A

Spośród poniższych zadań proszę zrobić przynajmniej dwa, w tym jedno spośród zadań 2 i 3.

Uwaga:

- Każdy plik powinien być podpisany w komentarzu na początku treści.
 - Proszę wysyłać również programy, których nie udało się Państwu dokończyć (z komentarzem w pliku, co już działa);
 - Każdy program powinien zawierać się w *jednym* pliku zawierającym wszystkie potrzebne definicje struktur danych i funkcji oraz testującą funkcję `main`. Nazwa pliku powinna być utworzona z nazwiska (bez odstępów i polskich znaków) i numeru zadania; np. `Kowalski2.cpp`. Jeśli potrzebny jest plik danych, to jego nazwa powinna być np. `Kowalski2.dat`.
 - Pliki proszę spakować i wysłać do p. dr Łopuch:
Bozena.Lopuch@ibspan.waw.pl
Nazwą wysłanego pliku-archiwum powinno być również nazwisko (z odpowiednim rozszerzeniem, np. `.zip`).
-

Zadanie 1

Napisz funkcję `bool Unikatowe(int A[], int n)`, która sprawdza, czy w nieposortowanej tablicy `A` o długości `n` ($n > 1$) znajdują się duplikaty. Funkcja zwraca `true`, jeśli nie ma duplikatów, w przeciwnym wypadku zwraca `false`. Napisz program (funkcję `main`) testujący tę funkcję.

Zadanie 2

Napisz funkcję, która zastępuje napis, poczynając od wskazanego miejsca do końca, nowym tekstem. Funkcja ma prototyp:

```
void replaceString(char *napis, char *zastap, int poczatek);
```

gdzie `napis` jest przetwarzanym ciągiem znaków, `zastap` nowym tekstem, `poczatek` jest indeksem pierwszego zastępowanego znaku w napisie `napis`. Nowe znaki powinny być wstawiane co najwyżej do osiągnięcia ostatniego znaku napisu `napis` (bo nie wiadomo, czy w pamięci jest miejsce na dodatkowe znaki!). Napisz program testujący funkcję.

Zadanie 3

Zdefiniuj strukturę `Ksiazka` o składowych

- `autor` — tablica znaków (o ustalonej długości, np. `char autor[70]`);
- `tytul` — podobnie;
- `liczbaEgz` — typu `int` (liczba egzemplarzy „na stanie”);
- `cena` — również typu `int`.

Program tworzy i przetwarza tablicę `ksiazki` obiektów typu `Ksiazka`. Należy:

1. wczytać dane książek i umieścić w tablicy `ksiazki` (funkcja `Wczytaj`);
2. wypisać zawartość tablicy `ksiazki` w czytelnej postaci (funkcja `Wypisz`);
3. wypisać dane najdroższej książki (funkcja `Najdrozsza`);
4. wypisać dane książki z najmniejszą liczbą egzemplarzy (funkcja `Najmniej`);
5. podać średnią cenę książki (funkcja `SredniaCena`).

Zadanie 4

Napisz program do obsługi wypożyczalni filmów. Program powinien umożliwiać przetwarzanie następujących informacji o poszczególnych filmach:

1. Tytuł filmu (napis);
2. Rok produkcji (liczba całkowita);
3. Rodzaj filmu (pojedyncza litera: **k**omedia, **d**ramat, **p**rzygodowy, **s**ensacja, **r**omans).

W pamięci te informacje powinny być przechowywane w postaci listy ze wskaźnikiem do początku listy i jej końca. Można wykorzystać deklaracje struktur danych i funkcję `DodajFilm` (dodającą kolejny element do listy) podane poniżej.

Opracowany program ma umożliwić użytkownikowi:

1. Tworzenie listy filmów;
2. Dodawanie nowego filmu do listy;
3. Zapisanie listy do pliku „Filmy.txt”;
4. Odczytanie listy z pliku „Filmy.txt”;
5. Usuwanie wskazanego filmu z listy;
6. Wyświetlenie informacji o filmach wskazanego przez użytkownika rodzaju.

Fim opis� wybraną przez siebie strukturą. Przykład:

```
const int MAX=30;
```

```
struct FILM {          struct WEZEL {          struct LISTA {
    char tytul[MAX];    FILM info;          WEZEL *pocz;
    int rok_prod;      WEZEL *nast;        WEZEL *kon;
    char rodzaj;      };
};
```

```
bool DodajFilm (LISTA &lista, FILM &info) {
    WEZEL *nowywezel = new WEZEL;
    if (!nowywezel)
        return false;
    nowywezel->info = info;
    nowywezel->nast = 0;
    if (lista.kon == 0)
        lista.pocz = lista.kon = nowywezel;
    else {
        (lista.kon)->nast = nowywezel;
        lista.kon = nowywezel;
    }
    return true;
}
```