

Uwaga: Należy rozwiązać cztery wybrane zadania.

Zadanie 1

Napisać 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 testujący opracowaną funkcję.

Zadanie 2

Zbuduj funkcję, która sprawdza czy jeden ciąg znaków występuje w drugim. Funkcja ma prototyp:

```
int findString(char *napis, char *szukam)
```

gdzie `napis` jest przeszukiwanym ciągiem znaków, `szukam` jest szukanym ciągiem znaków. Funkcja zwraca położenie szukanego ciągu w napisie lub `-1` jeśli poszukiwanie zakończy się niepowodzeniem. Napisz program testujący opracowaną funkcję.

Zadanie 3

W tablicy `Ksiazki[50]` przechowujemy dane książek sprzedawanych w naszej księgarni. Tablica zawiera następujące dane:

- `Autor[70]`,
- `Tytul[70]`,
- `liczbaEgz`,
- `Cena`.

Należy:

- wczytać dane książek i umieścić w tablicy `Ksiazki` (funkcja `Wczytaj`);
- wypisać zawartość tablicy `Ksiazki` w czytelnej postaci (funkcja `Wypisz`);
- wypisać dane najdroższej książki (funkcja `Najdrozsza`);
- wypisać dane książki z najmniejszą liczbą egzemplarzy (funkcja `Najmniej`),
- podać średnią cenę książki (funkcja `SredniaCena`).

W pierwszej dostawie do księgarni dostarczono cztery książki. Przetestować program dla zawartości księgarni po tej dostawie.

Zadanie 4

Dana jest lista struktur reprezentująca elementy rzadkiej tablicy dwuwymiarowej. Pojedyncza struktura danych tej listy zawiera trzy liczby (`w, k, v`) gdzie `w` – numer wiersza, `k` – numer kolumny, `v` – wartość elementu tablicy o współrzędnych (`w, k`). Liczby `w` i `k` są typu całkowitego bez znaku, a liczba `v` jest typu `double`.

Opracuj funkcję z argumentem w postaci odniesienia do listy, która zwraca liczbę kolumn. Napisz program testujący opracowaną funkcję.

Zadanie 5

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

- Tytuł filmu (`napis`),
- Rok produkcji (liczba całkowita),
- Rodzaj filmu (litera: **k**omedia, **d**ramat, **p**rzygodowy, **s**ensacja, **r**omans).

W pamięci komputera 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:

- Tworzenie listy filmów;
- Dodawanie nowego filmu do listy;
- Zapisanie listy do pliku „Filmy.txt”;
- Odczytanie listy z pliku „Filmy.txt”;
- Usuwanie wskazanego filmu z listy;
- Wyświetlenie informacji o filmach wskazanego przez użytkownika rodzaju.

Należy pamiętać o podpisaniu stworzonego przez siebie programu.

Fim opisać wybraną przez siebie strukturą.

<pre>const int MAX=30; struct FILM { char tytul[MAX]; int rok_prod; char rodzaj; }; struct WEZEL { FILM info; WEZEL *nast; }; struct LISTA { WEZEL *pocz; WEZEL *kon; } ;</pre>	<pre>struct FILM { char *tytul; int rok_prod; char rodzaj; }; struct WEZEL { FILM info; WEZEL *nast; }; struct LISTA { WEZEL *pocz; WEZEL *kon; } ;</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
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;
}
```