

Zadanie projektowe

Stworzyć obsługę struktury danych „stos”.

Wymagania.

1. Projekt powinien być wykonany w IDE VS 2022.
2. Projekt powinien zawierać:
 - a. Obsługę błędów i komunikatów. W projekcie nie powinno być fragmentów kodu takich jak:

```
FILE *pf = fopen("my_file.bin", "wb");  
if(!pf)  
    printf("Open File Error\n"); //Takich komunikatów będzie wiele i będą oni rozrzucane po  
//całej aplikacji.
```

Zamiast tego tu powinna być wywołana funkcja z obsługi błędów i komunikatów.
 - b. Obsługę stosu (Inicjowanie stosu, zwolnienie stosu, dodawanie nowego elementu, pobieranie pierwszego elementu, odnalezienie podanego elementu¹, zapis wszystkich elementów stosu na dysk w plik binarny, odczyt z dysku). Przy tym *obsługa stosu nie powinna zależeć od typu danych, umieszczonych w stosie. Oznacza to, że dowolna zmiana typu danych, przechowywanych w stosie, nie powinna zmienić żadnej linii kodu obsługi stosu.* Użyć dla tego wskaźnik `void *` dla przechowywanych danych oraz wywołanie funkcji obsługi danych przez wskaźniki do funkcji w obsłudze stosu. *Łączenie elementów w stos powinno odbywać się przez wskaźniki (Nie używać stosu, tworzego na podstawie tablicy!).*
 - c. Interfejs. Każda procedura z obsługi stosu powinna być wywołana z odpowiedniej funkcji interfejsu. W dodatek do tego interfejs powinien wyprowadzać na monitor elementy stosu oraz menu.
 - d. Dane. Powinni być przedstawione w postaci struktury i zawierać funkcji obsługi (zapis/odczyt pojedynczego obiektu na/z dysku, wyprowadzenie na monitor, inicjowanie obiektu oraz zwolnienie pamięci, funkcji porównywania według podanego kryterium przy poszukiwaniu i t. d.).
3. Każda składowa projektu (a, b, c, d) powinna być przedstawiona w postaci oddzielnej pary plików (cpp, h), przy czym dane powinni być zamknięte w pliku, a używane w innych plikach tylko poprzez funkcje. Wszystkie funkcje dla każdej ze składowych projektu powinni być rozdzielone na funkcje zewnętrzne (będą wykorzystane w innych plikach) i wewnętrzne (tylko w pliku, który obsługuje te dane). Funkcje wewnętrzne powinni być zamknięte w odpowiednich plikach.
4. Użyć strukturę danych `MY_STUDENT`, która zawiera nazwisko studenta (wiersz tekstowy), rok urodzenia (`int`) i kierunek studiów (element `enum`, który wskazuje do tablicy wierszy

¹ Za to, według jakiego kryterium będzie to odbywało, odpowiadają dane, przekazując wskaźnik do odpowiedniej funkcji danych.

tekstowych z nazwami poszczególnych kierunków). Wiersz tekstowy „nazwisko studenta” powinien być alokowany dynamicznie o rozmiarze, odpowiadającym ilości znaków w nazwisku.

5. Jako przykład wzorcowy można użyć zadanie Queue.

Zasady testowania przed wysłaniem do oceny.

1. Przed wysłaniem projektu jest **konieczne jego testowanie przez autora. Powinni być przetestowane wszystkie opcje menu**. Informuje, że jeśli przy uruchomieniu projektu okaże się, że aplikacja elementarne sypie się na GP (General Page Fault) albo opcja w ogóle nie działa, odkładam taki projekt do tyłu kolejki, i będzie taki projekt sprawdzany tylko po wyczerpaniu kolejki. Nie wykluczone, że to już będzie w II terminie. Jeśli to się powtórzy, anuluję zadanie i wydaję nowe, bardziej złożone.
2. Dla każdego zadania projektowego pozostał wprowadzony "licznik iteracji i ". Normalne zaliczenie – to $i \leq 3$. Jeśli $i > 3$, zastanawiam się o przyczynie. Jeśli dochodzę do wniosku, że przyczyna – zaniedbane zachowanie studenta w stosunku do uwag wykładowcy, odkładam taki projekt i będę jego sprawdzał po I terminie. Dzieje się to tak dla tego, żeby przerwać zabawę w ping-ponga: student szybciej coś poprawi i znów to wysyła. Sorry, to strasznie odbiera czas na sprawdzenie, i inne studenci nie dostają tego czasu.
3. **Proszę uważnie sprawdzić działanie zapisu-odczytu**. Dla tego wprowadzamy do stosu 3 elementy i dwa-trzy razy zapisujemy do pliku. Zamykamy aplikację, otwieramy znów i dwa-trzy razy odczytujemy dane z pliku. Wyświetlamy stos na monitorze i przekonujemy się, że mamy w RAM tylko jeden egzemplarz danych (nie powstało dublowania danych) i nie mamy *inwersji elementów stosu*. **Tylko po takim sprawdzeniu, plus do tego sprawdzenie wszystkich pozostałych opcji menu, mamy prawo wysłać projekt do oceniania.**