Przykład programu z kolokwium: **Jubiler**

Wyobraźmy sobie sklep jubilerski, w którym są sprzedawane kolczyki z diamentami. Każdy kolczyk ma jeden diament, każdy diament ma inną cenę. Wszystkie kolczyki mają taką samą oprawę, dlatego kupując parę kolczyków można wybrać dowolne dwa diamenty.
Należy napisać program gromadzący dane o asortymencie i podający cenę pary kolczyków (równą sumie cen dwóch wybranych diamentów - cena oprawy jest pomijalnie mała).
Żeby program nie był zbyt prosty – powinien mieć następującą strukturę:

- w jednej funkcji (gromadzącej dane o wszystkich diamentach) wczytywana jest liczba diamentów, a następnie ich ceny (każdy pojedynczy diament ma swoją indywidualną cenę),
- w drugiej funkcji (przetwarzającej wczytane dane do postaci gotowej do pobrania) są obliczane i zapamiętywane sumy każdej pary kolczyków (diamentów),
- w trzeciej funkcji (dla obsługi klienta) wczytywane są numery wybranych diamentów, a suma ich cen (zapamiętana, a nie jeszcze raz obliczana) jest wyprowadzana na ekran.

Zalecenia:

- Program, który nie korzysta z VLA (macierzy o „zmiennej" długości) będzie wyżej punktowany.
- Używanie zmiennych zewnętrznych zdecydowanie obniża punktowanie.
- Nie należy marnować pamięci – nie należy pamiętać dwukrotnie sumy tej samej pary kolczyków.
- Proponowana struktura danych (cen par): tablica wskaźników (w pamięci przydzielanej dynamicznie) do „coraz krótszych" tablic.
- Każda funkcja powinna zawierać komentarz – do czego służy, co oblicza, oraz opis parametrów.
- „Wcięcia" są obowiązkowe.
- Należy pamiętać o prawidłowej strukturze funkcji: W standardzie C90 definicje wszystkich zmiennych powinny być umieszczone na początku bloku.

Im więcej spełnionych zaleceń, tym lepszy wynik.

Jeżeli ktoś cierpi z powodu braku realizmu w tym programie („nie ma potrzeby pamiętania sum, bo w każdej chwili można ją policzyć!"), to proszę zastąpić ceny ciągami liczb rzeczywistych reprezentujących szeregi czasowe, a sumy cen – współczynnikami korelacji wzajemnej dla kilku przesunięć czasowych. Mam nadzieję, że w takim przypadku można bronić sensowności przedstawionej struktury programu. Żeby nie komplikować obliczeń – pozostańmy przy diamentach.

Program, który warto napisać w trakcie przygotowywania się do kolokwium:

Kolejka zleceń w tablicy (warto poszukać informacji o kolejkach, np. ostatni rozdział w Stephen Prata Szkoła oprogramowania – Język C: Zaawansowana reprezentacja danych – Kolejki)
Należy napisać program symulujący kolejkę zleceń (zlecenia są oznaczane kolejnymi liczbami naturalnymi). Przyjęte, a jeszcze nie obsłużone zlecenia oczekują w kolejce – są zapisane w **tablicy** zleceń. Zlecenia są obsługiwane w kolejności ich przyjmowania – (first in first out). Program powinien zawierać funkcje obsługujące kolejkę: dopisanie nowego zlecenia, wyrejestrowanie obsłużonego zlecenia, informowanie o przepełnieniu tablicy oraz o braku zleceń w kolejce.
Po każdej zmianie stanu kolejki program wyprowadza na ekran stan tablicy (bufora) zawierającej informacje o nie obsłużonych zleceniach. Puste miejsce w buforze jest zaznaczane znakiem _, np.: _ _ 3 4 5 _ _ _ _ _ oznacza, że tablica ma 10 elementów, w kolejce oczekują zlecenia nr 3, 4 i 5 (zlecenia 1 i 2 zostały już obsłużone)
Przebieg symulacji:

- wczytanie z klawiatury znaku I oznacza przyjęcie zlecenia (każde zlecenie otrzymuje kolejny numer – identyfikator),
- wczytanie z klawiatury znaku O oznacza obsługę zlecenia,
- po każdym znaku wczytanym pojawia się na ekranie ww. stan tablicy (bufora),
- wczytanie z klawiatury znaku X kończy program.

Wszelkie pytania odnoszące się do ww. programów (i inne) proszę kierować pod adres miller@agh.edu.pl