Programistyczne zadanie domowe nr III – tablice z haszowaniem

termin oddania: 2.06 (piątek, do końca dnia)

Samodzielnie zaimplementuj tablice z haszowanie w dowolnie wybranym języku. Przygotuj klasy/struktury implementujące tablice z haszowaniem wykorzystujące:

- metodę łańcuchową,
- adresowanie otwarte (liniowe, kwadratowe oraz haszowanie dwukrotne).

Konstruktor klasy (lub, w zależności od wybranego języka, coś będącego jego odpowiednikiem) powinien przyjmować:

- rozmiar tablicy m,
- rodzaj metody rozwiązywania kolizji (metoda łańcuchowa/rodzaj adresowania otwartego),
- funkcję haszującą (funkcje haszujące w przypadku haszowania dwukrotnego).

Dodatkowo klasa powinna posiadać następujące metody (w przypadku języka, który nie obsługuje klas mogą to być funkcje dodatkowo przyjmujące strukturę jako pierwszy argument):

- *insert(element)*,
- *delete(element)*,
- *search(element)*,
- *size()*.

Uwaga: Państwa kod nie może korzystać z gotowych bibliotek implementujących tablice z haszowaniem. Poza bibliotekami systemowymi dopuszczalne jest jedynie użycie bibliotecznej implementacji list (w przypadku metody łańcuchowej).

Wymagania dodatkowe

Państwa program powinien wczytywać dane z standardowego wejścia (*stdin*). Po uruchomieniu na standardowym wejściu pojawi się następujący ciąg znaków (poszczególne elementy są oddzielone od siebie spacjami):

$$rozmiar\ rodzaj\ element1\ element2\ element3\ ...$$

- $rozmiar$ — rozmiar tablicy (m),
- $rodzaj$ — rodzaj metody rozwiązywania kolizji (cyfra reprezentująca metodę: 1 – metoda łańcuchowa, 2 – adresowanie liniowe, 3 – adresowanie kwadratowe, 4 – haszowanie dwukrotne).
- $element1\ element2\ element3\ ...$ — lista elementów (do końca linii), które należy umieścić w tablicy (co najmniej jeden element). Elementy dodajemy do tablicy w porządku zgodnym z porządkiem ich przekazywania (tj. najpierw $element1$ potem $element2$ itd.).

Przykłady:

- $5\ 1\ 2\ 3$ —- 5-elementowa tablica z rozwiązywaniem kolizji za pomocą metody łańcuchowej. W tablicy zapisujemy liczby 2 i 3.
- $4\ 4\ 2\ 3\ 8\ 56\ 12\ 3$ —- 4-elementowa tablica z haszowaniem dwukrotnym. W tablicy zapisujemy liczby 2, 3, 8, 56, 12, 3.

Po wczytaniu danych, na standardowym wyjściu powinna być wyświetlona zawartość tablicy zapisana w następującej postaci:

$$[element, element, None, element, ...]$$

gdzie $element$ to wartość znajdująca się na danej pozycji (jeżeli jest nią lista, to wyświetlamy ją jako $[el1, el2, el3, ...]$). Wartość $None$ reprezentuje puste miejsce. Proszę zwrócić uwagę na spacje i dużą literę N w słowie $None$. Na wyjściu nie powinny się pojawiać żadne inne dane (teksty, etc.)!

Po wyświetleniu tablicy program przechodzi do trybu wstawiania/usuwania/wyszukiwania i przyjmuje na standardowym wejściu następujące komunikaty:

- -1 — zakończ działanie
- 0 $element$ — wstawianie nowego elementu $element$. Po wstawieniu wyświetlana jest cała zawartość tablicy (w taki sam sposób, jak po uruchomieniu programu),
- 1 $element$ — wyszukiwanie elementu $element$. W efekcie wywołania na standardowym wyjściu pojawia się pozycja elementu w tablicy (w przypadku metody łańcuchowej wyświetlamy jedynie pozycję całej listy w tablicy, bez podawania pozycji elementu w liście),
- 2 $element$ — usuwanie elementu $element$. Zakładamy, że usuwany element znajduje się w tablicy. Po usunięciu wyświetlana jest cała zawartość tablicy (w taki sam sposób, jak po uruchomieniu programu). W przypadku adresowania otwartego usunięta wartość powinna być reprezentowana w wyświetlanym tekście przez słowo $Deleted$ (a nie *None*).

W implementacji wykorzystaj funkcje haszujące ze zbioru zadań na ćwiczenia. W przypadku adresowania kwadratowego proszę zastosować $c_1 = c_2 = 1$.

W razie wątpliwości proszę o kontakt mailowy lub poprzez MS Teams.