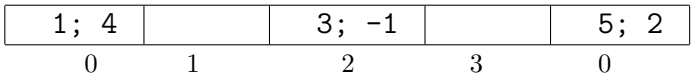
Compito di Programmazione
17 Febbraio 2022

Nota Bene. Ogni esercizio deve essere svolto su una pagina diversa. Alla fine, usando una applicazione che faccia la scansione, tipo CamScanner, fare una foto a tutto il compito col cellulare e inviare le scansioni in pdf a `cosimo.laneve@unibo.it`, `giuseppe.lisanti@unibo.it`, `adele.veschetti2@unibo.it`.

1. **(punti 8)** Un nodo è un elemento il cui tipo è `struct nodo { int val; int next ; } ;`.

   Una `lista_di_nodi` è implementata attraverso un array di elementi `nodo` la cui testa si trova ad indice 0 e in cui il campo `next` indica la posizione dell'elemento successivo. Se un nodo ha `next == -1` allora quel nodo è finale (non c'è alcun elemento successivo). Se un nodo ha `next == -2` allora quel nodo non fa parte della lista. Ad esempio, se il nodo a indice 0 ha il campo `next` uguale a -2 allora la lista è vuota.

   (a) Definire una funzione *ricorsiva* `there_is_space` che prende una `lista_di_nodi` e ritorna un'indice dell'array che non memorizza un nodo della lista. La funzione ritorna `-1` se l'array è pieno.

   (b) Definire una funzione `insert` che prende una `lista_di_nodi` e un intero e inserisce l'intero in coda alla lista (quando l'array non sia pieno).

2. **(punti 8)** Una libreria utilizza una lista per catalogare i libri che sono disponibili. Definire le strutture dati necessarie per rappresentare un catalogo di libri, tenendo presente che ogni libro è caratterizzato da un autore, il titolo del libro e il prezzo. Se ci sono più copie dello stesso libro, nel catalogo compare solamente una volta. Definire le seguenti funzioni:

   - `unisci_cataloghi` che prende come parametri due liste e ne crea una nuova contenente tutti gli elementi di entrambe le liste, ordinati per ordine alfabetico secondo l'autore del libro;
   - `piu_costoso` che ritorna il nome del libro più costoso tra quelli del catalogo.

3. **(punti 8)** Una `AutoOfficina` offre un insieme di servizi, ogni servizio è caratterizzato da un nome e un codice.

   (a) Si rappresenti il singolo servizio utilizzando una struttura dati.

   (b) Si implementi la classe `AutoOfficina`

   (c) il costruttore di `AutoOfficina`

   (d) si implementi `controlla_listino()`, il quale riceve il codice del servizio e controlla se quel servizio è presente. In caso positivo restituisce true, altrimenti restituisce false.

   (e) Quindi, si implementi la classe `AutoOfficinaPlus` la quale è caratterizzata, anche, da un insieme di materiali. Ciascun materiale è caratterizzato a sua volta da un codice e una quantità.

   - Si rappresenti il singolo materiale utilizzando una struttura dati.
   - Si implementi il costruttore di `AutoOfficinaPlus`;

- si implementi `effettua_servizio()`, il quale riceve il codice del servizio, due insiemi (uno contentente i codici dei materiali e l'altro la quantità necessaria di ciascuno di questi materiali) e la lunghezza di questi due insiemi. Il metodo controlla se il servizio è disponibile, in caso positivo decrementa la quantità dei materiali utilizzati, se disponibili. Il metodo restituisce `true` se il servizio può essere effettuato, altrimenti `false`.

Non è possibile utilizzare le liste.