

Sapi betina

▼ Mata Kuliah	Struktur Data
▼ Kelas	C
▼ Tahun Ajaran	2021/2022
▼ Praktikum Modul	1

Oleh: Luthfiyyah Hanifah Amari (5025201090)

Description

Pak imron ingin membeli sejumlah sapi betina dari Peternakan amari. sayangnya, uang beliau masih terbatas. imron membuat list sapi yang ada di peternakan tersebut. setiap sapi di sini mempunyai id. Pak imron menDAFTARKAN id sapi ke dalam listnya. list tersebut diurutkan berdasarkan prioritas pak imron dalam preferensi pak imron untuk membeli sapi betina. pak imron ingin memprioritaskan untuk membeli sapi betina yang lebih tua usianya. apabila ada usia sapi yang sama, maka imron memilih untuk memilih sapi yang lebih berat massanya. maka, sapi yang lebih diprioritaskan ditaruh di poin list yang lebih atas.

- dipastikan tidak ada 2 sapi yang memiliki usia dan massa yang sama

Peternakan ini terkadang mendapat pasokan sapi betina dari tempat lain. Apabila ada stock sapi baru, maka pak imron juga menDAFTARKAN id sapi itu ke dalam list (diurutkan sesuai dengan prioritasnya).

Sewaktu-waktu, pak Imron membeli sejumlah sapi. tentu saja sapi yang pak imron beli sesuai id sapi yang tertera pada list yang paling atas miliknya. apabila sapi dibeli, maka id sapi yang dibeli dihapus dari list. apabila ternyata jumlah sapi yang ingin dibeli pak imron lebih banyak daripada jumlah sapi yang ada di list, maka pak imron beli seadanya saja.

kamu diminta untuk membuat program yang menginput setiap pendaftaran sapi ke list dan pembelian sapi oleh pak imron. setelah mendapat semua perintah/query (daftar / beli), cetaklah id sapi yang tersisa pada list pak imron.

Input Format

baris pertama adalah N, banyaknya Query/perintah
N baris selanjutnya adalah Query yang dapat berupa

- `DAFTAR id x y` => artinya, seekor sapi didaftarkan dalam list. dimana, id ialah id sapi, x adalah usianya, dan y adalah massanya
- `BELI n` => artinya, pak imron membeli n sapi betina

Constraints

- $N < 20$
- $0 \leq x \leq 100$
- $0 \leq y \leq 1000$
- $n < 20$

Output Format

Mencetak id sapi yang tersisa pada list, berdasarkan urutan prioritas list

Sample Input 0

```
7
DAFTAR 1 90 500
BELI 3
DAFTAR 2 70 900
DAFTAR 3 70 950
DAFTAR 4 80 900
DAFTAR 5 80 940
BELI 1
```

Sample Output 0

```
4 3 2
```

Pembahasan

Memakai Priority Queue, tetapi fungsi pushnya dimodifikasi agar dapat menginsert data dengan 2 prioritas.

Source Code

```
/*  
  
- soal 2 modul 1 struktur data  
program ini menggunakan priority queue  
  
dibuat dan ditulis oleh luthfiyyah hanifah amari, masya Allah. alhamdulillah.  
--- tanggal 26 maret 2021.  
  
curhat dikit:  
    alhamdulillah buatnya lumayan lancar, masya Allah.  
  
catatan lain :  
    banyak source code yang copas dari source code modul github  
  
-----  
direvisi oleh luthfiyyah hanifah amari  
--- tanggal 3 maret 2022  
  
catatan: wkwk kok gini sih dulu buatnya?... but alhamdulillah sih, mungkin saya berkembang  
  
direvisi oleh luthfiyyah hanifah amari  
--- tanggal 5 maret 2022  
untuk penyelesaian soal praktikum modul 1  
kelas struktur data C 2022 genap  
  
*/  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <stdbool.h>
```

```

#include <string.h>
#define MAX_OUTPUT 30

// struct pqueue node
typedef struct pqueueNode_t {
    int data_urgensi1;
    int data_urgensi2;
    int data_id;
    struct pqueueNode_t *next;
} PQueueNode;

// struct priority queue
typedef struct pqueue_t {
    PQueueNode *_top;
    unsigned _size;
} PriorityQueue;

/* prototype fungsi */
void pqueue_init(PriorityQueue *pqueue);
bool pqueue_isEmpty(PriorityQueue *pqueue);
void pqueue_push(PriorityQueue *pqueue, int id, int urgensi1, int urgensi2);
void pqueue_pop(PriorityQueue *pqueue);
int pqueue_top(PriorityQueue *pqueue);

/* fungsi */

void pqueue_init(PriorityQueue *pqueue)
{
    pqueue->_top = NULL;
    pqueue->_size = 0;
}

bool pqueue_isEmpty(PriorityQueue *pqueue) {
    return (pqueue->_top == NULL);
}

int b;
void pqueue_push(PriorityQueue *pqueue, int id, int urgensi1, int urgensi2)
{
    PQueueNode *temp = pqueue->_top;
    PQueueNode *newNode = \
        (PQueueNode*) malloc (sizeof(PQueueNode));
    newNode->data_id = id;
    newNode->data_urgensi1 = urgensi1;
    newNode->data_urgensi2 = urgensi2;
    newNode->next = NULL;

    if (pqueue_isEmpty(pqueue)) {
        pqueue->_top = newNode;
        return;
    }

    if (urgensi1 > pqueue->_top->data_urgensi1)

```

```

    {
        newNode->next = pqueue->_top;
        pqueue->_top = newNode;
    }
    else if(urgensi1 == pqueue->_top->data_urgensi1 && urgensi2 > pqueue->_top->data_urgensi2)
    {
        newNode->next = pqueue->_top;
        pqueue->_top = newNode;
    }
    else {
        while ( temp->next != NULL &&
                temp->next->data_urgensi1 > urgensi1)
        {
            temp = temp->next;
        }

        if(temp->next != NULL && temp->next->data_urgensi1 == urgensi1)
        {
            while(temp->next != NULL &&
                  temp->next->data_urgensi2 > urgensi2 &&
                  temp->next->data_urgensi1 == urgensi1)
            {
                temp = temp->next;
            }
        }

        newNode->next = temp->next;
        temp->next = newNode;
    }
}

void pqueue_pop(PriorityQueue *pqueue)
{
    if (!pqueue_isEmpty(pqueue)) {
        PQueueNode *temp = pqueue->_top;
        pqueue->_top = pqueue->_top->next;
        free(temp);
    }
}

int pqueue_top(PriorityQueue *pqueue) {
    if (!pqueue_isEmpty(pqueue))
        return pqueue->_top->data_id;
    else return 0;
}

void pqueue_print(PriorityQueue *pqueue){
    PQueueNode *temp = pqueue->_top;
    while(temp != NULL){
        printf("%d ", temp->data_id);
        temp = temp->next;
    }
}

```

```

int main()
{
    char input[10];
    int temp_id, temp_urgensi1, temp_urgensi2;
    int output;
    PQueueNode daftar;
    int Q, i;

    pqueue_init(&daftar);

    scanf("%d", &Q);
    b = 0;

    for(i=0; i<Q; i++) {
        scanf("%s", &input);
        if(strcmp(input, "DAFTAR") == 0){
            scanf("%d %d %d", &temp_id, &temp_urgensi1, &temp_urgensi2);

            pqueue_push(&daftar, temp_id, temp_urgensi1, temp_urgensi2);
        }

        if(strcmp(input, "BELI") == 0){

            int n;
            scanf("%d", &n);
            int j;
            for(j=0; j<n; j++){
                pqueue_pop(&daftar);
            }
        }
    }

    while(!pqueue_isEmpty(&daftar)){
        printf("%d ", pqueue_top(&daftar));
        pqueue_pop(&daftar);
    }
    return 0;
}

```