Program memory: free; structs

COSC 208, Introduction to Computer Systems, 2021-09-29

Announcements

- Project 1 Part B (and revisions to Part A) due tomorrow at 11pm

Warm-up

Assume you wanted to write a function that creates a copy of a string. What is wrong with each of the following attempts at writing such a function?

Q1:

```
char *copy1(char strA[]) {
    char strB[strlen(strA) + 1];
    strcpy(strB, strA);
    return strB;
}
```

Q2:

```
char copy2(char strA[]) {
    char *strB = malloc(sizeof(char) * (strlen(strA) + 1));
    strcpy(strB, strA);
    return *strB;
}
```

Q3:

```
char *copy3(char strA[]) {
    char *strB = malloc(sizeof(char *));
    strcpy(strB, strA);
    return strB;
}
```

free

What memory deallocation mistake has been made in each of the following code snippets?

Q4:

```c
int *ptrA = malloc(sizeof(int) * 3);
int *ptrB = ptrA;
free(ptrA);
free(ptrB);
```

Q5:

```c
int *ptr = malloc(sizeof(int) * 3);
ptr[0] = 1;
free(ptr);
ptr[1] = 2;
```

Q6:

```c
int *ptr = malloc(sizeof(int) * 3);
ptr++;
free(ptr);
```

Q7:

```c
int *ptrA = malloc(sizeof(int) * 3);
int *ptrB = ptrA;
ptrA[0] = 0;
ptrB[1] = 1;
free(ptrA);
ptrB[2] = 2;
```

Pointers to structs

Assume you are given the following code:

```c
struct account {
    int number; // Account number
    int balance; // Current account balance
};
int deposit(struct account *acct, int amount);
int transfer(struct account *from, struct amount *to, int amount);
```

Q8: *Write the* `deposit` *function, which adds* `amount` *to the balance of* `acct`*. The function should return the amount deposited.*

Q9: *Write the* `transfer` *function which moves* `amount` *from one account to another. The function should return the amount transferred if the transfer was successful or 0 otherwise.*

Extra practice

Two structs have been defined representing a queue and an item on a queue.

```
struct item {
    int value;
    struct item *next;
};
struct queue {
    struct item *head;
    struct item *tail;
};
```

The new_queue function creates a new, empty queue.

```
struct queue *new_queue() {
    struct queue *q = malloc(sizeof(struct queue));
    q->head = NULL;
    q->tail = NULL;
    return q;
}
```

Q10: *Write a function called enqueue that adds a new value at the end of the queue.*

Q11: *Write a function called dequeue that removes and returns the value at the head of the queue. The function should return -1 if the queue is empty.*

Q12: *Write a function called free_queue that empties and frees a queue.*