

COP 3223 Homework #10: Charity Ball Organizer

Many charities support good causes, but one of the difficulties each of them has is organizing their fundraising events. After nearly a semester of C programming, you've decided that you'd like to donate your skills to create a program that organizes the typical activities at a fund-raising gala. In particular, your program will help manage the following:

- 1) Ticket sales
- 2) Silent Auction
- 3) Raffle
- 4) Drink sales

Your program will log the number of tickets sold both in advance and at the event, a silent auction for donated items, a raffle for other donated items and the number of drinks sold.

Ticket Sales Details

You will sell tickets in advance and at the door. Prices for buying in advance and at the door will be given. Also, the total number of tickets sold in advance will be given. Each guest will have a unique number. If there are n tickets sold in advance, then these guests will be numbered 0 through $n-1$. As the event starts, requests to buy tickets at the door may be made and these guests will be numbered sequentially, starting at the lowest unassigned number. The maximum number of guests will be 1000.

Silent Auction Details

The silent auction will have upto 1000 items, each with a given "market" value. At any time, users can bid on any item, so long as the bid exceeds the previous bid by the minimum increase value, which will be given. At the point in time when the auction is closed, the items with at least one bid are given to the users who have placed the last counted bid on that item. Your program must always keep track of all of the best bids on items so that no matter when the auction closes, you'll have all the data for who has won all of the items. Any item without a bid goes to no one. The market value of the item will be used to calculate each user's net "loss" or "gain".

Raffle Details

There will be a given number for the total number of raffle tickets sold, that will be 100000 or fewer. Guests can buy raffle tickets for a set price (this will be given). There will be a number of prizes awarded from the raffle (not to exceed 100) after the raffle has finished. For each of these prizes you will be given its "market" value. For each raffle ticket, you'll have to keep track of which guest has it. The raffle tickets will be numbered starting at 0 through the number of tickets minus one. After the raffle finishes, you will be given the raffle numbers pulled for the winners of each prize. You will be guaranteed that these numbers correspond to numbers that were previously handed out during the raffle. You must determine who wins each prize.

Drink Sales

There will be 10 kinds of drinks labeled 0 through 9. For each type of drink, you will be given an initial inventory and the cost of a single drink of that type. (You might be told that there are 100 of drink 0 each costing \$5.00, 50 of drink 1 each costing \$7.99, etc.) Guests may come up at any time and order any quantity of a particular drink. They will receive the number they request if that quantity is in inventory. Otherwise, they'll just receive what is left. If none of that particular type of drink is left, then they will receive no drinks. You will simply have to keep track of the total amount of money collected from drink sales, as well as the remaining inventory at the end of the night.

Input Details

In order to test your program thoroughly, you will be asked to take in input from a file and produce output for each event that occurs, in time.

Input File Format

The first line of the file contains the following three values, separated by spaces:

Cost of the presales tickets (in dollars), Cost of the tickets at the door (in dollars), and the number of presale tickets. The first two values will be positive real numbers to two decimal places and the last will be a positive integer.

The second line of the file will contain one positive integer representing the number of auction items followed by a positive real value to two decimal places (at most) representing the minimum bid increment, in dollars. The first value is guaranteed to be 1000 or less and the second will be in between 1 and 50, inclusive.

The third line of the file will contain all the prices of the auction items, in dollars, separated by spaces, in order. Thus, the first price is the price of item 0, the next price is the price of item 1, and so on. These values will be real numbers represented to upto 2 decimal places.

The fourth line of the file will contain the three following positive integers pertaining to the raffle: the number of raffle tickets available, the cost of a raffle ticket in dollars, and the number of raffle prizes. (It's strange to have raffle tickets that don't cost a whole number of dollars.)

The fifth line of the file will contain each of the values of the raffle items, in dollars, separated by spaces, in order. Thus, the first price is the price of item 0, the next price is the price of item 1, and so on. These values will be real numbers with upto 2 decimal places.

The sixth line of the file will contain ten positive integers representing the number of each of the drinks 0 through 9, in order, that are in stock.

The seventh line of the file will contain ten positive real numbers with upto 2 decimal places representing the price of each of the drinks 0 through 9, in order.

The eighth line of the file will contain a single positive integer, *numEvents*, representing the number of events that occur at the charity ball. These events are split into two groups: actions by guests at the ball and awards given (raffle, auction, person, totalrevenue). All of the actions precede all of the awards.

The following *numEvents* lines will contain information about each event that occurs, with one event or award described for each line.

You will produce exactly one line of output for each event described. Here are the formats of each event that could occur:

If a patron buys a ticket at the door, a command will be on a line by itself:

```
BUY TICKET k
```

where *k* is a positive integer indicating the number of tickets bought at the door. These guests will be numbered as previously mentioned. You are guaranteed that the total number of tickets bought, including presales, will not exceed 1000.

If a patron makes a bid in the silent auction, a command of the following form will be issued:

```
BIDITEM k p d
```

where *k* represents the item number for which the bid is being placed, *p* represents the number of the person placing the bid, and *d* represents the dollar amount the person has bid. This last value is a positive real number to two decimal places while the others are non-negative integers within the appropriate ranges.

The following command closes the auction. Any bids made after this command is executed are ignored.

```
CLOSEAUCTION
```

If a guest desires to buy raffle tickets, a command with the following format will be used:

```
BUY RAFFLE k p
```

This command lets person *p* buy exactly *k* raffle tickets. The raffle tickets are numbered as previously described, so long as the tickets are still available. If they aren't available, the person gets whatever tickets remain.

If a guest desires to buy drinks, a command with the following format will be used:

```
BUY DRINK k n
```

where k is the quantity of drinks being bought and n is the number of the drink, in between 0 and 9, inclusive, so long as that particular drink is in stock the sale is made. If fewer than k drinks are in stock, the remaining drinks are sold.

The last several commands in the file will deal with awarding prizes, and checking on the status of individuals. These commands will take the following forms.

A query about a raffle item will be of the following form:

```
AWARD RAFFLE i t
```

where i is the raffle item being given out and t is the number of the winning ticket of that raffle item. Your program will have to respond with the winner of this item. You will be guaranteed that this query will appear EXACTLY once for each raffle prize and that all queries of this form will appear AFTER the last raffle ticket is sold. Furthermore, you are guaranteed that the raffle ticket number was purchased by an individual and that the raffle prize number is valid. Finally, a single raffle ticket will correspond to at most one raffle prize.

A status query about a silent auction item will be of the following form:

```
AWARD AUCTION i
```

where i represents the silent auction item being queried. Your program will have to respond with the winner of this item (you are guaranteed that all queries will be about items that have at least one bid, and hence, a winner) and the amount they paid for it. You will receive exactly one query on each silent auction item that received one or more bids.

A status query about an individual will be of the following form:

```
AWARD PERSON i
```

For this query, you'll simply have to determine person i 's net gain or loss at the event. Their total gain is the sum of the values of raffle items and silent auction items they have been awarded. Their total cost is the price of the ticket they bought for the event (you can determine if it was an advance ticket or not), the cost of all the raffle tickets they bought, and the amount they spent on each item they won in the silent auction. Note that drink costs are not factored in here because it's assumed that the value of the drink is equal to its cost. A person's net gain/loss is their total gain minus their total cost.

The final status query in the file will be the following line:

```
TOTAL REVENUE
```

Output Format

For the input command of the form:

```
BUY TICKET k
```

output a single line of the form:

```
SOLD TICKETS a - b
```

where a is the starting number of the tickets sold and b is the ending number (inclusive) of the tickets sold. If k is 1, then a and b will be equal. (Note: For the very first command of this type, the value of a will equal the number of presale tickets, since the presale tickets are numbered from 0 to the total number of presale tickets minus 1.)

For the input command of the form:

```
BIDITEM k p d
```

output a single line with one of the following forms:

```
BIDITEM k ACCEPTED for PERSON p at d DOLLARS  
BIDITEM k REJECTED for PERSON p at d DOLLARS
```

The two reasons to reject a bid are if the raise of the bid isn't equal to or more than the minimum required or if the auction has been closed already. For an initial bid, it must be equal or more than the minimum raise of the bid. Thus, we assume that without any bids, the first price of the item is \$0.

For the CLOSEAUCTION command, simply print a single line with the exact same output.

For the input command

```
BUY RAFFLE k p
```

output a single line of one of the two following forms:

```
RAFFLE TICKETS a - b given to PERSON p  
NO RAFFLE TICKETS given to PERSON p
```

where a represents the first raffle ticket number issued and b represents the last raffle ticket number issued to person p. If there are no more raffle tickets left with the command is issued, the latter format is used. Note that the former format may be used if the person asks for a certain number of raffle tickets, but gets fewer since she bought all that were left.

For the input command

```
BUY DRINK k n
```

output a single line of one of the two following forms:

```
SOLD k of DRINK n  
NO DRINK n SOLD
```

where k represents the number of the drinks sold (a positive integer) and n represents the drink number, ranging from 0 through 9, inclusive. Note that for this line of output, the value of k may differ from the number of drinks requested by the guest, if the inventory runs out. If the drink requested is already sold out, the latter message should be printed.

For the input command

```
AWARD RAFFLE i t
```

output a single line of the following format:

```
RAFFLE i WON BY PERSON p
```

where i is the raffle item in question (in the query) and p is the number of the guest who won the item.

For the input command

```
AWARD AUCTION i
```

output a single line of the following format:

```
AUCTION ITEM i WON BY PERSON p for $d.dd
```

where i is the number of the auction item, p is the person who won the item and d.dd is the number of dollars he paid for it, to two decimal places.

For the input command

```
AWARD PERSON i
```

output a single line of one of the two following formats:

```
PERSON i is UP $d.dd  
PERSON i is DOWN $d.dd
```

If the person is exactly even, you may use either format. Otherwise, pick the appropriate format based on whether or not they've gained or lost value from the event accordingly. You may not receive a query for each person who attended the auction.

For the input command

```
TOTALREVENUE
```

output a single line with the following format:

```
TOTALREVENUE is $d.dd
```

where d.dd is the total amount gained by the event via ticket sales, auction sales, raffle ticket sales and drink sales. (We are assuming that all items were donated so that there was no cost associated with the event.)

Sample Input /Output and Grading Criteria

Several files will be posted with this assignment. These files are indicative of the files on which your program will be tested. The execution points you'll receive for these types of cases will be given to you. (Each case will have increasing complexity and worth more points.) Remember that documentation and programming style will still be worth 40% of the total program grade. This is a very, very challenging assignment, so there is no shame in not completing this assignment. The important part is that you solve at least some parts of it. Try to add features to your program to solve each test case given, in order.

Deliverables

One source file: *charityball.c*, for your solution to the given problem.

All files are to be submitted over WebCourses. **(Do NOT submit .cpp files!!!)**

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Please use Code::Blocks to develop your program. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your program will be graded upon the following criteria:

- 1) Your correctness.
- 2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.
- 3) A reasonable program design made manageable with several functions.
- 4) Compatibility with Code::Blocks (in Windows). If your program does not compile in this environment, you will get a **sizable** deduction from your grade.