CSC180 Introduction to Computer Programming
Assignment 1

Due: 11:59 PM, October 6, 2008

1 Problem description

Simulate the receipt machine at the U of T Bookstore. Prices are only allowed to be an integral number of cents. You must define the functions in Section 2 using exact function prototypes. Hint: use global variables. This assignment is worth 4% toward your final grade.

2 Function prototypes

- `void init(char c);`

  Initialize the receipt machine for a customer. This function sets the number of items purchased by the customer to 0, subtotal price (price before tax) to 0, and tax type to `c`, where `c` can be `'g'` or `'G'` (GST), `'p'` or `'P'` (PST), `'b'` or `'B'` (both GST and PST), or any other character (tax free). Let GST be 5%, and PST be 8%.

- `void purchase(int q, double p);`

  Purchase `q` items, each with price `p` (in CA dollars).

- `void cancel(int q, double p);`

  Cancel `q` items, each with price `p`. `cancel()` would work even if you try to cancel items that haven't been bought yet. However, note that `cancel()` must not make the number of items purchased negative, nor make subtotal price negative. (See Section 4 for more information.) Any attempts to violate this requirement will have no effects.

- `double tax(char c);`

  Return tax imposed on customer's effective purchases, given tax type `c`. Purchases that are not cancelled are considered effective.

- `double total(void);`

  Return the total amount of money that the customer needs to pay (including tax, if any).

- `double change_due(double a);`

  Return change due given payment `a`.

- `double avg(void);`

  Return average cost of an item after tax.

- `void print(double a);`

  Given payment `a`, print a U of T Bookstore receipt, in the following format:

```
U of T Bookstore Receipt
#item: 15
Subtotal: $3.00
Tax (PST + GST): $0.39
Total: $3.39
Payment: $20.00
Change due: $16.61
```

  , where `#item` is the number of effective items purchased.

  The above receipt was generated by the following function calls:

```
init('b');          /* set tax type */
purchase(10, 0.1);  /* buy 10 items, each $0.1 */
purchase(20, 0.2);  /* buy 20 items, each $0.2 */
cancel(5, 0.2);     /* cancel 5 items, each $0.2 */
cancel(10, 0.1);    /* cancel 10 items, each $0.1 */
print(20);          /* pay $20, and print receipt */
```

  `print()` is the only function in this assignment that prints information on standard output. All other functions must not print anything.

3 Receipt format

Failure to satisfy the following requirements will result in considerable mark deductions.

- The receipt must have exactly the same format as the one shown above, except that the numbers may vary. `#item` must be a non-negative integer, with no decimal point nor fraction part. Money must be preceded by `$`, and must have two and only two digits after decimal point.

- All field names (characters before :) must be present, in the exact form, and in the order as shown above.

- There must be one and only one space after colon.
- Only the following tax field names are acceptable:

```
Tax (PST)
Tax (GST)
Tax (PST + GST)
Tax (none)
```

- If no tax was imposed, the tax line prints:

```
Tax (none): $0.00
```

- No spaces or tabs after each line

4 Handling round-off errors

Floating-point numbers of type `double` are not exact. Allowing any double to be a price leads to unavoidable problems with round-off errors. For example, if a customer purchases 20 items one by one each costing $1.31, and then cancels them all at once, he gets a negative number. If a customer purchases 99 items at the same price and cancel them one by one, he again gets a negative number. In such situations, cancellation can still be made. Failure to handle these situations will result in losing 5% of the total mark for this assignment.

5 Submission

- We provide the header file `receipt.h`; its contents are listed below. You must use this file, and must not modify it.

```
/* receipt.h */

/* macro definitions */
#define GST  0.05
#define PST  0.08

/* function prototypes */
void init(char);
void purchase(int, double);
void cancel(int, double);
double tax(char);
double total(void);
double avg(void);
```

```
double change_due(double);
void print(double);
```

- You must submit one and only one file, `receipt.c`. Include the following line near the top of `receipt.c`:

  ```
  #include ‘‘receipt.h’’
  ```

  `receipt.c` must not contain `main` function, nor other header files except `stdio.h` and `math.h`. You may test the functions defined in `receipt.c` using another file, e.g., `main.c`. `main.c` may look like this:

  ```
  #include <stdio.h>
  #include ‘‘receipt.h’’

  int main()
  {
    /* test functions from receipt.c */
    return 0;
  }
  ```

- Insert the following comment on the top of `receipt.c`:

  ```
  /*
   * receipt.c -- description of this file, if any
   *
   * $name, 012345678$
   */
  ```

  Replace `name` with your name, and replace `012345678` with your student number.

- Comment `receipt.c` properly
- Compilation: compile the program using the following command:

  ```
  gcc -Wall receipt.c main.c -o receipt
  ```

- Submission methods: to be announced.