

```

1  /*
2  *****
3  * JaMe CPE Coffee Shop System.
4  * Ask each customer for a member code (1 Character,5 digits), if not this is a
non-member If the customer enters " -1 ",
5  * program will show the end of the day and print the summary member ,non-member
,income ,discount and reduced income by discount .
6  * Firstly, program will ask for the type of coffee i.e. Espresso($30), Latte($40),
Cappuccino($45), Cocoa($35),
7  * the size that is Small or Large(It will charge double of price), temperature
namely Hot or Cold(it will charge 15%).
8  * Finally, program will ask amount of coffee that you bought.
9  *
10 * Members get a 30% discount on all purchases. When the customer enters the word
"NONE" for drink type will print a bill for
11 * the table showing each drink ordered, and show price of that drink before
discounts, and the final price including discounts.
12 *
13 * Developed by Siwat Karwlung(JaMe) ID 56070503434
14 * Date 27/9/2556
15 * Latest update 10/10/2556 9:04 A.M.
16 *****
17 */
18 #include <stdio.h> /*printf fgets*/
19 #include <string.h> /*strcpy strlen*/
20 #include <stdlib.h> /*memset calloc malloc realloc*/
21 #include <stdbool.h> /*boolean type is 1 bit use less memory*/
22 #include <ctype.h> /*isalpha*/
23 #include <time.h> /*time_t, struct tm, time, localtime*/
24 #include <limits.h> /*USHRT_MAX*/
25
26 #define MAX_LEN 32 /*max length of buffer*/
27 #define LOOP_TIME 64 /*to clear screen by newline feed 64 time*/
28
29 #define MAX_COFFEE 100 /*max coffee order time for 1 customer*/
30 /*****
31 * This function clear stdin in standard C
32 * input - input from fgets
33 * no return
34 *****/
35 */
36
37 /*void clearStdin(char *input)
38 {
39     printf("Please enter to continue...\n");
40     if(input[MAX_LEN-2] != '\n' && input[MAX_LEN-1] == '\0')
41     {
42         while (getchar() != '\n');
43     }
44 }
45 */
46 /*****
47 * This function clear screen by for-loop and newline-feed
48 * no input
49 * no return
50 *****/
51 */
52
53 void clearScreen()
54 {
55     unsigned short i = 0;
56     for(i=0;i<LOOP_TIME;i++)
57     {
58         printf("%c", '\n');/*newline*/
59     }
60     printf("%c", '\r');/*set cursor to beginning of line*/

```

```

61     }
62
63     /*****
64     * This function check input to character
65     * input - input string
66     * return true if input is character
67     *     false if input is not character
68     *****/
69     */
70     bool bCharacter(char *input)
71     {
72         unsigned short i;
73         for(i=0;i<strlen(input);i++) /*use loop to check each character if not
alphabetic it will return false*/
74         {
75             if(!isalpha(input[i])) return false;
76         }
77         return true;
78     }
79     /*****
80     * This function check input to numeric
81     * input - input string
82     * return true if input is character
83     *     false if input is not character
84     *****/
85     */
86     bool bNumeric(char *input)
87     {
88         unsigned short i;
89         for(i=0;i<strlen(input);i++)/*use loop to check each character if not numeric it
will return false*/
90         {
91             if(!isdigit(input[i])) return false;
92         }
93         return true;
94     }
95     /*****
96     * This function ask user a kind of coffee
97     * if user enter input which is not coffee
98     * it will repeat loop to ask until it correct
99     * it always convert input to lower case to check to result
100    * return price of coffee
101    *****/
102    */
103    unsigned short checkCoffee()
104    {
105        char input[MAX_LEN];/*to store text from user*/
106        memset(input,'\0',sizeof(input));/*set text to null*/
107        printf("What would you like sir ?\n");/*ask user coffee*/
108        printf("Espresso ($30), Latte ($40), Cappuccino ($45), Cocoa ($35)\n,(NONE to
stop buying coffee)\n");
109        fgets(input,sizeof(input),stdin);/*get input from standard input*/
110        sscanf(input,"%[^\\n]",input);/*get everything from input except \\n(new line)*/
111        /*clearStdin(input);*/
112        unsigned short i;
113        for(i=0;i<strlen(input);i++)/*use loop to check if each character is upper case
it will convert to lower case*/
114        {
115            if(isupper(input[i]))
116            {
117                input[i] = tolower(input[i]);
118            }
119        }
120        while((strcmp("espresso",input) != 0) && (strcmp("latte",input) != 0) && (strcmp
("cappuccino",input) != 0) && (strcmp("cocoa",input) != 0) && strcmp("none",input) != 0)
121            /*accordingly we have lower case string so we use while loop if user not

```

```

enter coffee type or each type of coffee are maximum cup it will repeat ask until correct
*/
122     printf("Please enter only coffee name !(example: cocoa)\n");/*error wrong
input not coffee type*/
123     printf("What would you like sir ?\n");
124     printf("Espresso ($30), Latte ($40), Cappucino ($45), Cocoa ($35)\n,(NONE to
stop buying coffee)\n");
125     fgets(input,sizeof(input),stdin);
126     sscanf(input,"%[^\\n]",input);
127     /*clearStdin(input);*/
128     for(i=0;i<strlen(input);i++)/*use loop to check if each character is upper
case it will convert to lower case*/
129     {
130         if(isupper(input[i]))
131         {
132             input[i] = tolower(input[i]);
133         }
134     }
135     /*it will return price of coffee if user enter correct coffee*/
136     if(strcmp("espresso",input) == 0) return 30;
137     else if(strcmp("latte",input) == 0) return 40;
138     else if(strcmp("cappucino",input) == 0) return 45;
139     else if(strcmp("cocoa",input) == 0) return 35;
140     else if(strcmp("none",input) == 0) return 0;
141     else
142     {
143     }
144 }
145
146 /*****
147 * This function ask user temperature of coffee
148 * if user enter input which is not temperature
149 * it will repeat loop to ask until it correct
150 * it always convert input to lower case to check to result
151 * return false - hot
152 *         true - cold
153 *****/
154 */
155 bool checkTemp()
156 {
157     char input[MAX_LEN];/*to store text from user*/
158     memset(input,'\\0',sizeof(input));/*set text to null*/
159     printf("What would temperature which you prefer to (Hot or Cold[Charge 15%])
?\\n");/*ask user temperature*/
160     fgets(input,sizeof(input),stdin);/*get input from standard input*/
161     sscanf(input,"%[^\\n]",input);/*get everything from input except \\n(new line)*/
162     /*clearStdin(input);*/
163     unsigned short i;
164     for(i=0;i<strlen(input);i++)/*use loop to check if each character is upper case
it will convert to lower case*/
165     {
166         if(isupper(input[i]))
167         {
168             input[i] = tolower(input[i]);
169         }
170     }
171     while(!(strcmp("hot",input) == 0 || strcmp("cold",input) == 0))
172     {/*accordingly we have lower case string so we use while loop if user not
enter temperature it will repeat ask until correct*/
173         printf("Please enter hot or cold only !(example : hot)\\n");
174         printf("What would temperature which you prefer to (Hot or Cold[Charge
15%]) ?\\n");
175         fgets(input,sizeof(input),stdin);
176         sscanf(input,"%[^\\n]",input);
177         /*clearStdin(input);*/
178         for(i=0;i<strlen(input);i++)

```

```

179         {
180             if(isupper(input[i]))
181             {
182                 input[i] = tolower(input[i]);
183             }
184             /*it will return false when hot & true when cold*/
185         }
186         if(strcmp("hot",input) == 0) return false;
187         else if(strcmp("cold",input) == 0) return true;
188         else
189         {
190             }
191     }
192     /*****
193     * This function ask user a size of coffee
194     * if user enter input which is not size
195     * it will repeat loop to ask until it correct
196     * it always convert input to lower case to check to result
197     * return 1 - small size
198     *      2 - big size
199     *****/
200     /*
201     unsigned short checkSize()
202     {
203         char input[MAX_LEN];/*to store text from user*/
204         memset(input, '\0', sizeof(input));/*set text to null*/
205         printf("What would size which you prefer to (Small or Large[x2 price]) ?\n");
206         /*ask user size*/
207         fgets(input, sizeof(input), stdin);/*get input from standard input*/
208         sscanf(input, "%[^\\n]", input);/*get everything from input except \\n(new line)*/
209         /*clearStdin(input);*/
210         unsigned short i;
211         for(i=0; i<strlen(input); i++)/*use loop to check if each character is upper case
212         it will convert to lower case*/
213         {
214             if(isupper(input[i]))
215             {
216                 input[i] = tolower(input[i]);
217             }
218             /*accordingly we have lower case string so we use while loop if user not
219             enter size it will repeat ask until correct*/
220             while(!(strcmp("big",input) == 0 || strcmp("small",input) == 0 || strcmp("large",
221             input) == 0 || strcmp("tiny",input) == 0))
222             {
223                 printf("Please enter small or large only !(example : small)\n");
224                 printf("What would size which you prefer to (Small or Large[x2 price]) ?\n"
225                 );
226                 fgets(input, sizeof(input), stdin);
227                 sscanf(input, "%[^\\n]", input);
228                 /*clearStdin(input);*/
229                 for(i=0; i<strlen(input); i++)
230                 {
231                     if(isupper(input[i]))
232                     {
233                         input[i] = tolower(input[i]);
234                     }
235                 }
236                 /*it will return 1 when small size & 2 when big*/
237             }
238             if(strcmp("small",input) == 0) return 1;
239             else if(strcmp("tiny",input) == 0) return 1;
240             else if(strcmp("large",input) == 0) return 2;
241             else if(strcmp("big",input) == 0) return 2;
242             else
243             {
244             }
245         }

```

```

240     }
241     /*****
242     * This function ask user a amount of coffee
243     * if user enter input which is amount
244     * it will repeat loop to ask until it correct
245     * it always convert input to lower case to check to result
246     * return amount of coffee
247     *****/
248     */
249     unsigned short checkAmount()
250     {
251         char input[MAX_LEN]; /*to store text from user*/
252         unsigned short convert; /*to store amount*/
253         memset(input, '\0', sizeof(input)); /*set text to null*/
254         printf("What would you want amount of it ? :"); /*ask user amount*/
255         fgets(input, sizeof(input), stdin); /*get input from standard input*/
256         sscanf(input, "%[^\n]", input); /*get everything from input except \n(new line)*/
257         sscanf(input, "%hu", &convert); /*covert string to short*/
258         /*clearStdin(input);*/
259         while(bNumeric(input) == false || convert < 1) /*use loop to check input is
numeric if not it will repeat ask until correct*/
260         {
261             printf("What would you want amount of it ?(Numeric 1-65535 only):");
262             fgets(input, sizeof(input), stdin);
263             sscanf(input, "%[^\n]", input);
264             sscanf(input, "%hu", &convert); /*covert string to short*/
265             /*clearStdin(input);*/
266         }
267         return convert; /*return amount of coffee*/
268     }
269     /*****
270     * This function ask user member code if user enter -1 it means end of the day
271     * return -1 - to stop and show report end of the day
272     * 0 - if user enter incorrect member code
273     * 1 - if user enter correct member code(example:A29144)
274     *****/
275     */
276     short checkMember()
277     {
278         char input[MAX_LEN]; /*to store text from user +1 for if player enter more than 6
character*/
279         memset(input, '\0', sizeof(input)); /*set text to null*/
280         printf("Enter you member code[1 character, 5 digits] (\\"-1\\" to end of the
day)\n(Members get a 30%% discount on all purchases):"); /*ask user member code*/
281
282         fgets(input, sizeof(input), stdin); /*get input from standard input*/
283         sscanf(input, "%[^\n]", input); /*get everything from input except \n(new line)*/
284         /*clearStdin(input);*/
285         if(strcmp(input, "-1") == 0) /*if input is -1 it will return -1 to end of the
day*/
286         {
287             return -1;
288         }
289         if(strlen(input) != 6 || bNumeric(&input[1]) == false || isalpha(input[0]) == 0)
/*to check member if user input correct input, this condition will return false (not
member)*/
290         { /*non member*/
291             printf("You are not member.\n");
292             return 0;
293         }
294         printf("You are member.\n");
295         return 1; /*return 1 is member*/
296     }
297
298     /*****
299     * This function is to printf result of the end

```

```

300  * of the day
301  * Argument : member - amount of member
302  *           : nonMember - amount of non-member
303  *           : totalCharge - sum of coffee price
304  *           before discount
305  *           : totalDiscount - sum of discount
306  *           coffee price
307  * No return
308  ****
309  */
310
311  void printEndoftheday(unsigned short member,unsigned short nonMember,double
totalCharge,double totalDiscount)
312  {
313      time_t rawtime;/*variable to show current time*/
314      struct tm * timeinfo;
315      time (&rawtime);/*clear screen*/
316      timeinfo = localtime (&rawtime);
317      clearScreen();/*clear screen*/
318      printf("End of the day !\n",asctime(timeinfo));
319      clearScreen();/*clear screen*/
320      printf("%s","\a\a\a");/*beep sound 3 times*/
321      printf("JaMe's coffee shop report ,Date %s\n",asctime(timeinfo));/*show current
time*/
322      printf("*****\n");
323      printf("Member : %hu person\n",member);/*show the number of member*/
324      printf("Non-member : %hu person\n",nonMember);/*show the number of non-member*/
325      printf("Total charge of coffee : $%.2lf\n",totalCharge);/*show charge of
coffee*/
326      printf("Total discount of coffee : $%.2lf\n",totalDiscount);/*show discount of
coffeee*/
327      printf("Grand total :$%.2lf\n",totalCharge-totalDiscount);/*show grand total*/
328      printf("*****\n");
329  }
330
331  /*****
332  * This function is to bill for 1 customer
333  * Argument : coffeeTime - how many time that user bought coffee
334  *           : member - amount of member
335  *           : nonMember - amount of non-member
336  *           : totalCharge - sum of coffee price
337  *           before discount
338  *           : totalDiscount - sum of discount
339  *           coffee price
340  * No return
341  ****
342  */
343
344  void printBill(unsigned short coffeeTime,unsigned short *coffeeCustomer,unsigned
short *sizeCustomer,unsigned short *temperatureCustomer,unsigned short *amountCustomer,
double *priceCoffee,double *priceCoffeeDiscount,double customerCharge,double
customerDiscount)
345  {
346      unsigned short j;/*to order and show the result coffee*/
347      char *coffeeDatabase[] = {"Espresso","Latte","Cappucino","Cocoa"};/*for show
coffee in string by subscript*/
348      char *temperatureDatabase[] = {"Hot","Cold"};/*for show temperature in string by
subscript*/
349      char *sizeDatabase[] = {"Small","Large"};/*for show size in string by
subscript*/
350      time_t rawtime;/*variable to show current time*/
351      struct tm * timeinfo;
352      time (&rawtime);
353      timeinfo = localtime (&rawtime);
354      printf("Customer bill, Date %s\n",asctime(timeinfo));/*show current time*/
355      printf("*****\n");

```

```

356      /*show each coffee cup, price before discount,discount price and price after
discount*/
357      for(j=0;j<coffeeTime;j++)
358      {
359          printf("Drink type %s :\tSize: %s\tTemperature: %s\tAmount: %hu cup\nFull
price: $%.2lf\tDiscount price: $%.2lf\nPrice after discount: $%.2lf\n\n"
360              ,coffeeDatabase[coffeeCustomer[j]],sizeDatabase[sizeCustomer[j]],
temperatureDatabase[temperatureCustomer[j]],amountCustomer[j],priceCoffee[j],
priceCoffeeDiscount[j],priceCoffee[j]-priceCoffeeDiscount[j]);
361      }
362      printf("*****\n");
363      /*show total price with discount,discount price and price after discount*/
364      printf("Grand total without discount : $%.2lf\n",customerCharge);
365      printf("Total discount :$%.2lf\n",customerDiscount);
366      printf("Final total:$%.2lf\n",customerCharge-customerDiscount);
367      }
368
369      /*****
370      * This function is mainly to do everything such as call another function
371      * give a result such as bill and report
372      * count member and non-member
373      *****/
374      /*
375      void askMemberCoffee()
376      {
377          unsigned short coffeeCustomer[MAX_COFFEE],temperatureCustomer[MAX_COFFEE],
sizeCustomer[MAX_COFFEE],amountCustomer[MAX_COFFEE];/*each ordered information of each
coffee for 1 customer*/
378          double priceCoffee[MAX_COFFEE],priceCoffeeDiscount[MAX_COFFEE]; /* to store
price before discount and price after discount each time*/
379          unsigned short i=0;/*to order each coffee from 1 customer & to show order coffee
respectively*/
380          double customerCharge = 0,customerDiscount = 0; /*sum of expense of each
customer and discount */
381          double totalCharge = 0,totalDiscount = 0; /*result end of the day*/
382          unsigned short member = 0,nonMember = 0; /*count member and non-member*/
383          short bDiscount = 0; /*status to check member to calculate discount price*/
384          float temperature = 1; /*status to check temperature to calcualte price*/
385          unsigned short coffee = 0; /* store price coffee to calculate*/
386          while((bDiscount = checkMember()) != -1)/* ask member code until member type
"-1" to show result end of the day*/
387          {
388              if(bDiscount == 1)
389              {
390                  member++;/*count member*/
391              }
392              else
393              {
394                  nonMember++;/*count non-member*/
395              }
396              while((coffee = checkCoffee()) != 0 && i < MAX_COFFEE)/*ask member a coffee
type until type "NONE" to show bill*/
397              {
398                  sizeCustomer[i] = checkSize();/*ask member a size of coffee*/
399                  temperatureCustomer[i] = checkTemp();/*ask member a temperature of
coffee*/
400                  while((amountCustomer[i] = checkAmount()) > USHRT_MAX)
401                  {
402                      /*this loop will ask user amount until coffect input*/
403                      printf("You can enter amount less than or equal to %d from your
buying.\n",USHRT_MAX);
404                      amountCustomer[i] = checkAmount();
405                  }
406                  if(coffee == 30) coffeeCustomer[i] = 0;/*in function coffee,it will
return the price, so we need to store value in subscript to show in string because we
have coffeeDatabase in printBill function*/
407                  else if(coffee == 40) coffeeCustomer[i] = 1;

```

```

407         else if(coffee = 45) coffeeCustomer[i] = 2;
408         else if(coffee = 35) coffeeCustomer[i] = 3;
409         else { }
410         if(temperatureCustomer[i] == true)
411             {
412                 temperature = 1.15;/*if cold charge 15%*/
413             }
414         else
415             {
416                 temperature = 1.0;/*if hot no charge*/
417             }
418         if(bDiscount == 1)/*if member, store discount each coffee discount
price*/
419             {
420
421                 priceCoffeeDiscount[i] += coffee*sizeCustomer[i]*amountCustomer[i]*
0.3*temperature;
422                 customerDiscount += coffee*sizeCustomer[i]*amountCustomer[i]*0.3*
temperature;/*store sum of price discount*/
423             }
424         else
425             {
426                 priceCoffeeDiscount[i] = 0.0;
427             }
428         priceCoffee[i] += coffee*sizeCustomer[i]*amountCustomer[i]*temperature;
/*store each coffee price before discount*/
429         customerCharge += coffee*sizeCustomer[i]*amountCustomer[i]*temperature;
/*store sum of price before discount*/
430         sizeCustomer[i] --;/* generally,it is 2 or 1,we use 2,1 for calculate
buy subscript to show string start at 0 ,so decreasing for using in show table for
subscript the sizeDatabase array*/
431         i++;/*count coffee in each time*/
432     }
433     printBill(i,coffeeCustomer,sizeCustomer,temperatureCustomer,amountCustomer,
priceCoffee,priceCoffeeDiscount,customerCharge,customerDiscount);/*printf bill for 1
customer*/
434     totalCharge += customerCharge;/*add customer price to sum of charge of the
day*/
435     totalDiscount += customerDiscount;/*add customer discount price to sum of
discount of the day*/
436     customerCharge = 0; customerDiscount = 0;/*Reset all variable to get new
customer*/
437     memset(coffeeCustomer,0,sizeof(coffeeCustomer)); memset(sizeCustomer,0,
sizeof(sizeCustomer));
438     memset(temperatureCustomer,0,sizeof(temperatureCustomer)); memset(
amountCustomer,0,sizeof(amountCustomer));
439     i=0;
440 }
441 printEndoftheday(member,nonMember,totalCharge,totalDiscount);
442 }
443
444 void main()
445 {
446     clearScreen();/*clear screen*/
447     printf("Welcome To JaMe Coffee Shop\n\n\n\n\n\n\n\n\n\n");
448     askMemberCoffee();
449 }

```