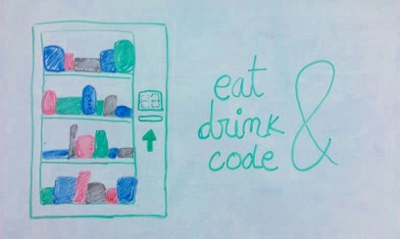
**Tema 2**

**eat & drink & code**

[](http://cursuri.cs.pub.ro/~programare/w/index.php/Imagine:Eat-drink-code-small.jpg)

**Deadline**: luni, 21 noiembrie 2011, ora 23:55

**Responsabil**

* Victor Cărbune [vic...@gmail.com](http://www.google.com/recaptcha/mailhide/d?k=013-Z3Yy3LgZyKZOFOHIA3iA==&c=gRETqRdOi8MiLN1Ow4ASgMbY5ecL-Y8JZ9abey9mC28=)
* Laura Vasilescu [lau...@cti.pub.ro](http://www.google.com/recaptcha/mailhide/d?k=013-Z3Yy3LgZyKZOFOHIA3iA==&c=gqCAKSTDk8nQKO2D3T0ePPT_fhDGlQajQiW9TjhLqjE=)

**Obiective**

În urma realizării aceste teme, studentul va fi capabil să:

* lucreze cu vectori
* lucreze modularizat
* implementeze funcții conform unei specificații date
* înțeleagă utilitatea și modul de folosire a testelor unitare

**Anunturi**

**[21/11/2011 20:35]** Actualizat arhiva [checker](http://swarm.cs.pub.ro/%7Evictor/checker.zip).

**[12/11/2011 17:07]** Actualizat arhiva [checker](http://swarm.cs.pub.ro/%7Evictor/checker.zip).

**[12/11/2011 16:52]** Actualizat descrierea force\_request\_product.

**[12/11/2011 16:17]** In urma [intrebarilor](http://cs.curs.pub.ro/2011/mod/forum/discuss.php?d=447#p2507), am actualizat descrierea functiei request\_product din antet.

|  |
| --- |
| **Cuprins**  [[ascunde](javascript:toggleToc())]   * [1 Introducere](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_2#Introducere) * [2 Cerință](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_2#Cerin.C8.9B.C4.83) * [3 Implementare](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_2#Implementare) * [4 Upload](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_2#Upload) * [5 Punctaj și specificații](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Tema_2#Punctaj_.C8.99i_specifica.C8.9Bii) |

**Introducere**

Încă din momentul în care au devenit colegi de apartament, conviețuirea dintre Leonard și Sheldon s-a bazat pe un contract de apartament, întocmit și redactat cu atenție de Sheldon.

În ultimele luni Leonard, datorită unor modificări în cadrul laboratorului din universitate, este nevoit să lucreze un timp semnificativ de acasă. Sheldon, fiind foarte atent la cantitatea de snacks-uri și sucuri pe care le deține din frigiderul comun, a sesizat că Leonard are obiceiul de a consuma și a nu le înlocui.

Cum acest aspect nu este prevăzut în contractul celor doi colegi de apartament, Sheldon petrece câteva nopți pentru a adăuga secțiunea corespunzătoare: un **automat de snacks-uri și sucuri** programabil, astfel încât Leonard, împreună cu musafirii săi, să achite corespunzător consumul făcut.

Întrucât singurul care are skill-uri de engineering (dintre cercul lor restrâns de prieteni) este Howard, Leonard a reușit să ajungă la o înțelegere cu acesta.

**Cerință**

Task-ul vostru este să îl ajutați pe Howard să implementeze acest automat.

Howard a primit de la Sheldon specificațiile cu care automatul trebuie să funcționeze și le-a scris într-un fișier anter (.h), pentru limbajul C.

*/\*\*  
 \* Homework 2 - C Programming  
 \* Author: ?  
 \* License: ?  
 \*/*  
   
#ifndef \_\_AUTOMAT\_H  
#define \_\_AUTOMAT\_H  
   
#define MAX\_NUMBER\_OF\_PRODUCTS 100  
   
*/\*\*  
 \* Tipurile de bancnote disponibile.  
 \*/*  
typedef **enum** {  
 \_1\_DOLLAR = 1,  
 \_5\_DOLLARS = 5,  
 \_10\_DOLLARS = 10,  
 \_50\_DOLLARS = 50  
} bill\_type;  
   
*/\*\*  
 \* Seteaza numarul de bancnote disponibile din tipul dat.  
 \*/*  
void set\_available\_bills(bill\_type bill, int count);  
   
*/\*\*  
 \* Intoarce numarul de bancnote disponibile de un anumit tip.  
 \* (-1, daca tipul respectiv de bancnote nu exista).  
 \*/*  
int get\_available\_bills(bill\_type bill);  
   
*/\*\*  
 \* Seteaza numarul de produse disponibile. Initial, acest  
 \* numar trebuie sa fie 0.  
 \*/*  
void set\_number\_of\_products(int count);  
   
*/\*\*  
 \* Intoarce numarul total de produse disponibile.  
 \* Este vorba de numarul total de produse, indiferent de cantitate.  
 \*/*  
int get\_number\_of\_products();  
   
*/\*\*  
 \* Intoarce costul unui anumit produs, dat prin codul sau.  
 \* Fiecare produs este identificat printr-un cod unic.  
 \*/*  
int get\_product\_cost(int code);  
   
*/\*\*  
 \* Seteaza costul unui produs.  
 \*/*  
void set\_product\_cost(int code, int cost);  
   
*/\*\*  
 \* Seteaza cantitatea disponibila a unui anumit produs.  
 \*/*  
void set\_product\_quantity(int code, int quantity);  
   
*/\*\*  
 \* Intoarce cantitatea disponibila a unui anumit produs.  
 \*/*  
int get\_product\_quantity(int code);  
   
*/\*\*  
 \* Cere un produs.  
 \*  
 \* Urmatoarele coduri de eroare trebuie intoarse, in diferite situatii:  
 \* -1 -- a fost cerut un produs care nu exista (cod invalid)  
 \* -2 -- produsul nu mai este disponibil (cantitate = 0)  
 \* -3 -- insuficienti bani introdusi pentru a cumpara produsul.  
 \* -4 -- imposibil de returnat rest cu bancnotele disponibile.  
 \*   
 \* Daca nu se intoarce un cod de eroare, request\_product va intoarce  
 \* suma totala pe care ar putea sa o ofere drept rest, in cazul in care  
 \* request\_change este apelata imediat dupa aceea. Intern, bancnotele  
 \* disponibile nu vor fi modificate pana la apelul request\_change.  
 \*/*  
int request\_product(int code);  
   
*/\*\*  
 \* Cere restul.  
 \*  
 \* Nota: acest automat este special, in sensul ca un client poate sa  
 \* nu ceara intotdeauna rest, sau sa uite!  
 \*  
 \* Daca functia este apelata dupa request\_product (\*doar daca\* cumparatorul  
 \* doreste sa faca asta) i se va intoarce restul disponibil.  
 \*  
 \* Daca request\_product nu a fost apelat si au fost bani introdusi,  
 \* aceasta functie va intoarce aceeasi suma, in orice bancnote  
 \* disponibile.  
 \*  
 \* Daca force\_request\_product este apelata, cumparatorul nu doreste  
 \* rest, si aceasta functie nu ar trebui apelata.  
 \*/*  
int request\_change();  
   
*/\*\*  
 \* Cere un produs, chiar daca nu exista rest disponibil.  
 \* Codurile intoarse sunt aceleasi ca si la request\_product,  
 \* cu exceptia erorii in privinta restului.  
 \*/*  
int force\_request\_product(int code);  
   
*/\*\*  
 \* Insereaza o bancnota.  
 \* (-1, daca bancnota nu a putut fi identificata).  
 \*/*  
int insert\_bill(bill\_type bill);  
   
*/\*\*  
 \* Reseteaza automatul (reset ar trebui sa reseteze intreaga stare,  
 \* numar de produs, cantitati, costuri, etc).  
 \*/*  
void reset();  
   
#endif /\* \_\_AUTOMAT\_H \*/

**Implementare**

În cadrul acestei teme veți implementa funcționalitățile descrise în fișierul antet mai sus, într-un fișier sursă denumit **automat.c**.

Pe lângă **automat.h** și **automat.c**, tema va mai conține și un fișier sursă principal **tema2.c**, cu o funcție main() care va simula practic comportamentul unui automat.

În cadrul **tema2.c** trebuie să folosiți toate funcțiile implementate, împreună cu mesaje corespunzătoare de intrare și ieșire. Un inceput ar putea fi:

int main() {  
 int produse;  
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("Configurati numarul de produse disponibile in automat:");  
   
 [scanf](http://www.opengroup.org/onlinepubs/009695399/functions/scanf.html)("%d", &produse);  
 set\_number\_of\_products(produse);  
   
 [printf](http://www.opengroup.org/onlinepubs/009695399/functions/printf.html)("Sunt %d produse disponibile: %d", get\_number\_of\_products());  
 (...)  
}

Precum un automat, ./tema2 ar trebui sa aștepte introducerea unor bancnote și cererea unui produs, respectiv servirea acestuia. Apreciem creativitatea :)

Testarea se va face automat, prin intermediul unor [teste unitare](http://en.wikipedia.org/wiki/Unit_testing) specifice fiecărei funcții.

Este foarte important ca în automat.c să implementați toate funcțiile din automat.h (checker-ul nu verifică niciun aspect al fișierului tema2.c).

**Upload**

Tema se va trimite conform regulilor de trimitere precizate în [regulament](http://cursuri.cs.pub.ro/%7Eprogramare/w/index.php/Regulament_general#Reguli_de_trimitere_a_temelor).

În curând o să fie disponibil upload-ul pe vmchecker.

**Punctaj și specificații**

* 90 puncte: trecerea testelor unitare (vmchecker, [checker](http://swarm.cs.pub.ro/%7Evictor/checker.zip))
* 10 puncte: simularea automatului în tema2.c, coding style și README