#include <stdio.h>

#include <stdlib.h>

typedef struct Cookie{

char name[30];

float price;

int num;

}product;

typedef struct Order{

char adr[30];

int number;

}order;

int main()

{

/\* Zadacha 1

int table=0,chair=0,cups=0,dishes=0,guests;

printf("Number of guests: ");

scanf("%d",&guests);

while (1){

char \*str = (char\*)malloc(sizeof(char));

scanf("%s",str);

if (strcmp(str,"PARTY!")==0){

break;

}

else{

if (strcmp(str,"Table")==0){

table++;

}

if (strcmp(str,"Chair")==0){

chair++;

}

if (strcmp(str,"Cups")==0){

cups++;

}

if (strcmp(str,"Dishes")==0){

dishes++;

}

}

free(str);

}

float sum=table\*42+chair\*13.99+cups\*5.98+dishes\*21.02;

printf("Sum =%.2f\n",sum);

if (guests/8>table){

printf("%d Table\n",guests/8-table);

}

if (guests/8==table && guests%8>0){

printf("%d Table\n",guests/8-table+1);

}

if (guests>chair){

printf("%d Chairs\n",guests-chair);

}

if (guests/6>cups){

printf("%d Cups\n",guests-cups);

}

if (guests/6==cups && guests%6>0){

printf("%d Cups\n",guests-cups+1);

}

if (guests/6>dishes){

printf("%d Dishes\n",guests/6-dishes);

}

if (guests/6==dishes && guests%6>0){

printf("%d Dishes\n",guests/6-dishes+1);

}

Zadacha 2

product \*p=(product\*)malloc(sizeof(product));

order \*m=(order\*)malloc(sizeof(order));

int a=0,b=0;

while (1){

printf("Command: ");

char str[10];

scanf("%s",&str);

if (strcmp(str,"END")==0){

break;

}

else if (strcmp(str,"Product")==0){

printf("Name: ");

scanf("%s",&p[a].name);

printf("Price: ");

scanf("%f",&p[a].price);

printf("Number: ");

scanf("%d",&p[a].num);

for (int i=0;i<b;i++){

if (p[a].num==m[i].number){

printf("Client %s ordered %s \n",m[i].adr,p[a].name);

m[i].number=-1;

}

}

a++;

realloc(p,sizeof(p)+1);

}

else if (strcmp(str,"Order")==0){

printf("Adress: ");

scanf("%s",&m[b].adr);

printf("Number: ");

scanf("%d",&m[b].number);

for (int i=0;i<a;i++){

if (p[i].num==m[b].number){

printf("Client %s ordered %s \n",m[b].adr,p[i].name);

m[b].number=-1;

}

}

b++;

realloc(m,sizeof(m)+1);

}

}\*/

return 0;

}