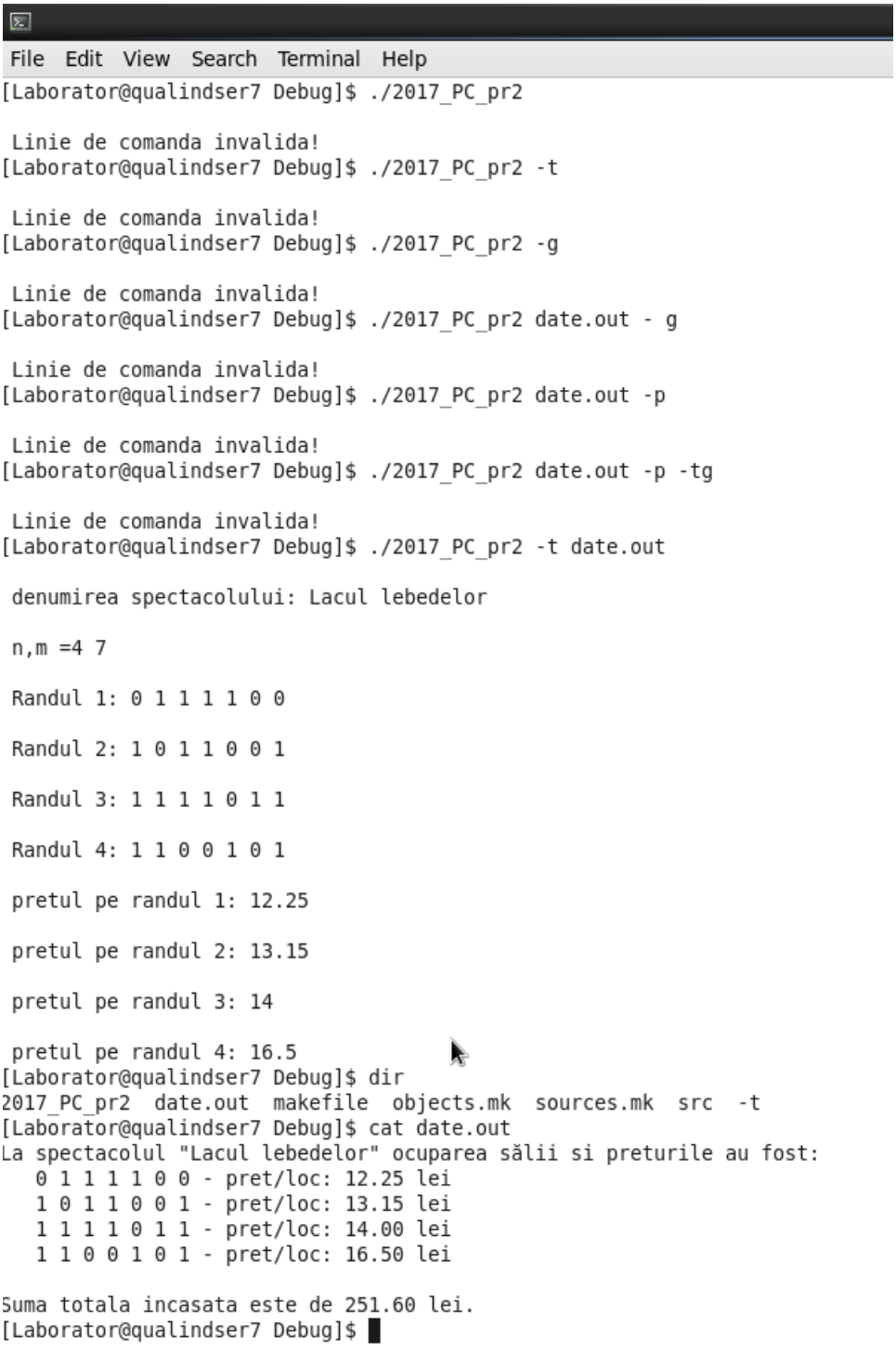
Rezolvare model de subiect practic P2

Programarea Calculatoarelor sesiunea iunie 2017



header.h

```
#ifndef HEADER_H_
#define HEADER_H_

void* xmalloc(size_t nrOcteti);
int** aloca2d(size_t n, size_t m);
void dealoca2d(int** a, size_t n);
int** citireMatrice(size_t n, size_t m);
double* citireVector(size_t n);
void afisare1(char* den, int** a, double* p, size_t n, size_t m, FILE* f);

double prelucrare_p(int** a, size_t n, size_t m);
double prelucrare_t(int** a, double* p, size_t n, size_t m);
int prelucrare_r(int** a, double* p, size_t n, size_t m);

double valRand(int** a, double* p, size_t r, size_t m);

typedef double (*pfp)(int** a, size_t n, size_t m);
typedef double (*pft)(int** a, double* p, size_t n, size_t m);
typedef int (*pfr)(int** a, double* p, size_t n, size_t m);

void afisare2(int** a, double* p, size_t n, size_t m,char flag, FILE* f);
#endif /* HEADER_H_ */
```

functii.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "header.h"

void* xmalloc(size_t nrOcteti)
{
	void* p=0;
	p=malloc(nrOcteti);
	if(!p)
	{
		fprintf(stderr,"Memorie insuficienta!");
		exit(EXIT_FAILURE);
	}
	return p;
}


int** aloca2d(size_t n, size_t m)
{
	int** a=0, i;
	a=(int**)xmalloc(n*sizeof(int*));
	for(i=0;i<n;i++)
		a[i]=(int*)xmalloc(m*sizeof(int));
	return a;
}


void dealoca2d(int** a, size_t n)
{
	int i;
	for(i=0;i<n;i++)
	{
		free(a[i]);
		a[i]=NULL;
	}
	free(a);
}


int** citireMatrice(size_t n, size_t m)
{
	int** a=0;
	int i,j;
	a=aloca2d(n,m);
	for(i=0;i<n;i++)
	{
		printf("\n Randul %d: ", i+1);
		for(j=0;j<m;j++)
		{
			scanf("%d",&a[i][j]);
		}
	}
	return a;
}
```

```c
double* citireVector(size_t n)
{
	int i;
	double* v=0;
	v=(double*)xmalloc(n*sizeof(double));
	for(i=0;i<n;i++)
	{
		printf("\n pretul pe randul %d: ", i+1);
		scanf("%lf",&v[i]);
	}
	return v;
}


void afisare1(char* den, int** a, double* p, size_t n, size_t m, FILE* f)
{
	int i,j;
	fprintf(f,"La spectacolul \"%s\" ocuparea sălii si preturile au fost:\n", den);
	for(i=0;i<n;i++)
	{
		fprintf(f,"   ");
		for(j=0;j<m;j++)
		{
			fprintf(f,"%d ", a[i][j]);
		}
		fprintf(f,"- pret/loc: %.2lf lei \n", p[i]);
	}
}

double prelucrare_p(int** a, size_t n, size_t m)
{
	double rez=0;
	int i,j;
	for(i=0;i<n;i++)
		for(j=0;j<m;j++)
			rez+=a[i][j];
	return rez/(n*m)*100;
}


double prelucrare_t(int** a, double* p, size_t n, size_t m)
{
	double rez=0;
	int i, j;
	double sum;
	for(i=0;i<n;i++)
	{
		sum = 0;
		for(j=0;j<m;j++)
			sum+=a[i][j];
		rez+=sum*p[i];
	}
	return rez;
}
```

```
int prelucrare_r(int** a, double* p, size_t n, size_t m)
{
	double max= valRand(a,p,0,m), aux;
	int i, poz=0;
	for(i=1;i<n;i++)
	{
		aux=valRand(a,p,i,m);
		if(max<aux)
		{
			max=aux;
			poz=i;
		}
	}
	return poz;
}

void afisare2(int** a, double* p, size_t n, size_t m,char flag, FILE* f)
{
	pfp pp;
	pfr pr;
	pft pt;

	switch(flag)
	{
	case 'p':
		pp = &prelucrare_p;
		fprintf(f,"\nProcentul de ocupare a salii a fost de %.3lf%%.\n",(*pp)(a,n,m));
		break;
	case 'r':
		pr = &prelucrare_r;
		fprintf(f,"\nRandul cu încasarea cea mai mare este randul %d cu %.2lf lei.\n",
(*pr)(a,p,n,m)+1, valRand(a,p,(*pr)(a,p,n,m),m));
		break;
	case 't':
		pt = &prelucrare_t;
		fprintf(f,"\nSuma totala incasata este de %.2lf lei.\n",(*pt)(a,p,n,m));
		break;
	default:
		break;
	}

}

double valRand(int** a, double* p, size_t r, size_t m)
{
	double rez=0;
	int i;
	for(i=0;i<m;i++)
		rez+=a[r][i];
	return rez*p[r];
}
```

Main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "header.h"

int main(int argc, char* argv[])
{
	char buff[100], *den, *numeFis;
	int** a;
	double* p;
	size_t n, m;
	FILE* f;
	int iflag = 0;
	char flag;


	switch(argc)
	{
	case 1:
		iflag = 1;
		break;
	case 2:
		if(argv[1][0]=='-')
			iflag = 1;
		else
		{
			flag = 'p';
			numeFis=(char*)xmalloc((strlen(argv[1])+1)*sizeof(char));
			strcpy(numeFis, argv[1]);
		}
		break;
	case 3:
		if(argv[1][0]!='-')
			iflag = 1;
		else
		{
			switch(argv[1][1])
			{
			case 'p':
				flag='p';
				break;
			case 'r':
				flag='r';
				break;
			case 't':
				flag='t';
				break;
			default:
				iflag = 1;
				break;
			}
			if(!iflag)
			{
				numeFis=(char*)xmalloc((strlen(argv[2])+1)*sizeof(char));
				strcpy(numeFis, argv[2]);
```

```
                        }
                }
                break;
        default:
                iflag = 1;
                break;
        }



        if(iflag)
                printf("\n Linie de comanda invalida!\n");
        else
        {
                printf("\n denumirea spectacolului: ");
                fgets(buff, 99, stdin);
                buff[strlen(buff)-1]='\0';
                den =(char*)xmalloc((strlen(buff)+1)*sizeof(char));
                strcpy(den, buff);

                printf("\n n,m =");
                scanf("%u %u",&n,&m);

                a=citireMatrice(n,m);

                p=citireVector(n);

                f = fopen(numeFis,"w");
                if(!f)
                {
                        fprintf(stderr,"Eroare la deschiderea fisierului");
                        exit(EXIT_FAILURE);
                }
                afisare1(den, a, p, n, m, f);
                afisare2(a,p,n,m,flag,f);

                fclose(f);

                dealoca2d(a,n);
                a=NULL;
                free(p);
                p=NULL;
        }
        return 0;
}
```