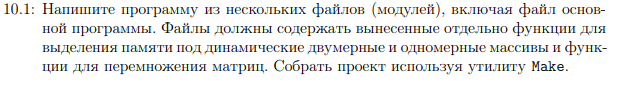
Лабораторная работа №9 «Процессор, модульная сборка программы»

Постановка задачи:



**memory.c**

#include <stdio.h>

#include <stdlib.h>

int \*\*createm(int n,char AA[]) {

int i,j;

int \*\*M = (int\*\*)malloc(n\*sizeof(int\*));

for (i=0;i<n;i++)

M[i] = (int\*)malloc(n\*sizeof(int));

for (i=0;i<n;i++)

for (j=0;j<n;j++) {

printf("%s[%d][%d]: ",AA,i+1,j+1);

scanf("%d",&M[i][j]);

}

return M;

}

int\* createa(int n,char AA[]) {

int i;

int \*A = (int\*)malloc(n\*sizeof(int));

for (i=0;i<n;i++) {

printf("%s[%d]: ",AA,i+1);

scanf("%d",&A[i]);

}

return A;

}

**memory.h**

#ifdef MEMORY\_H

#define MEMORY\_H

int\*\* createm(int, char);

int\* createa(int, char);

#endif // MEMORY\_H

**product.c**

#include <stdio.h>

#include <stdlib.h>

int\*\* prod(int\*\* Am,int\*\* G,int n, char AA[]) {

int i,j,k;

int \*\*M =(int\*\*)malloc(n\*sizeof(int\*));

for (i=0;i<n;i++)

M[i] = (int\*)malloc(n\*sizeof(int));

for (i=0;i<n;i++)

for (j=0;j<n;j++) {

M[i][j] = 0;

for (k=0;k<n;k++)

M[i][j] += Am[i][k]\*G[k][j];

}

return M;

}

**product.h**

#ifdef PRODUCT\_H

#define PRODUCT\_H

int\*\* prod(int\*\*, int\*\*, int,char);

#endif // PRODUCT\_H

**freedome.c**

#include <stdio.h>

#include <stdlib.h>

void freed(int \*\*M,int n) {

int i;

for (i=0;i<n;i++)

free(M[i]);

free(M);

}

void fre(int \*A) {

free(A);

}

**freedome.h**

#ifdef FREEDOME\_H

#define FREEDOME\_H

void freed(int\*\*, int)

void fre(int\*)

#endif // FREEDOME\_H

**main.c**

#include <stdio.h>

#include <stdlib.h>

#include "memory.h"

#include "product.h"

#include "freedome.h"

int main(void) {

int n,i,j,m;

printf("Enter is n: ");

scanf("%d",&n);

int \*\*A = createm(n,"A");

printf("\n");

int \*\*B = createm(n,"B");

printf("\n");

int \*\*AB = prod(A,B,n,"AB");

printf("A\*B:\n");

for (i=0;i<n;i++) {

for (j=0;j<n;j++)

printf("%d\t",AB[i][j]);

printf("\n");

}

printf("Enter size of array: ");

scanf("%d",&m);

printf("one-dimensional array:\n");

int \*a = createa(m,"a");

for (i=0;i<m;i++)

printf("%d\n",a[i]+10);

freed(A,n);

freed(B,n);

fre(a);

}

**makefile**

CC=gcc

CFLAGS=-std=c17

DEPENDENCIES=main.c product.h memory.h freedome.h

OBJECT\_DEPENDENCIES=main.o product.o memory.o freedome.o

LIBRARIES=-lm

.DEFAULT\_GOAL=main

%.o: %.c $(DEPENDENCIES)

$(CC) $(CFLAGS) -c -o $@ $<

main:$(OBJECT\_DEPENDENCIES)

$(CC) $(CFLAGS) -o $@ $^ $(LIBRARIES)

Результат:

