1.

#include <unistd.h>

#include <stdio.h>

#include <pthread.h>

#include <errno.h>

#include <stdlib.h>

#include <string.h>

void\* reverse(void \*sir){

char \*str = (char\*) sir;

char \*final=malloc(strlen(str)+1);

int i, j=0;

for(i=strlen(str)-1;i>=0;i--){

final[j] = str[i];

j+=1;

}

final[j]=0;

return final;

}

int main(int argc, char \*argv[]){

char \*result;

pthread\_t thr;

if(pthread\_create(&thr, NULL, reverse, argv[1])){

perror(NULL);

return errno;

}

if(pthread\_join(thr, &result)){

perror(NULL);

return errno;

}

printf("%s\n", result);

free(result);

}

2.

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <pthread.h>

#define rows 2

#define columns 2

int m1[rows][columns]={

{1,2},

{3,4}

};

int m2[rows][columns]={

{5,6},

{7,8}

};

int rez[rows][columns]={0};

struct pos{

int i,j;

};

void \*inmultire(void \*poz){

struct pos\* pozi = poz;

int i = pozi->i;

int j = pozi->j;

free(poz);

for(int k=0; k<columns; k++)

rez[i][j]=rez[i][j] + m1[i][k]\*m2[k][j];

return NULL;

}

int main(){

pthread\_t thr[rows\*columns];

int thr\_id=0;

for(int i=0;i<rows;i++)

for(int j=0;j<columns;j++){

struct pos\* pozi=calloc(1, sizeof(struct pos));

pozi->i=i;

pozi->j=j;

if(pthread\_create(&thr[thr\_id++], NULL, inmultire, pozi)){

perror(NULL);

return errno;

}

for(int i=0;i<thr\_id;i++)

if(pthread\_join(thr[i], NULL)){

perror(NULL);

return errno;

}

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

for(int j=0;j<columns;++j)

printf("%d ", rez[i][j]);

printf("\n");

}

return 0;

}