//copiere linii pare

#include <stdlib.h>#include <stdio.h>#include <unistd.h>#include <string.h>#include <fcntl.h>#include <sys/stat.h>#include <sys/types.h>

void copiere (char \*fisierInput, char\* fisierOutput){

int descriptor1 = open (fisierInput, O\_RDONLY);

if (descriptor1 < 0) {

puts ("Nu am gasit fisierul de intrare");

return;

}

int descriptor2 = open (fisierOutput, O\_WRONLY | O\_CREAT, S\_IRUSR | S\_IWUSR);

char ch;

int par = 0;

while (read (descriptor1, &ch, 1) == 1){

if (par == 1) {

write (descriptor2, &ch, 1);

}

if (ch == '\n') {

par = 1 - par;

}

}

close (descriptor1);

close (descriptor2);

}

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

if (argc != 3){

puts ("Numar invalid de argumente");

return -1;

}

copiere (argv[1], argv[2]);

return 0;

}

//fibonacci parerel

#include <stdlib.h>;#include <string.h>;#include <ctype.h>;#include <unistd.h>;#include <fcntl.h>;#include <sys/types.h>;#include <sys/stat.h>;#include <sys/wait.h>;#include <sys/sem.h>

void fibonacci (int nr){

int i;

for (i = 1; i <= nr; i++)

{

if (fork() == 0){

if (i == 1 || i == 2) {

exit (1);

}

int a, b, c;

a = 1, b = 1, c;

int j;

for (j = 3; j <= i; j++){

c = a + b;

a = b;

b = c;

}

exit (c);

}

}

int status;

while (wait(&status) > 0){

if (WIFEXITED(status) != 0) {

printf ("%d ", WEXITSTATUS(status));

}

}

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

}

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

if (argc != 2) {

puts ("Numar invalid de argumente");

return -1;

}

int nr; // numarul primit ca argument

sscanf (argv[1], "%d", &nr);

fibonacci (nr);

return 0;

}

//fibonacci secvential 1

#include <stdlib.h>;#include <stdio.h>;#include <string.h>;#include <ctype.h>;#include <unistd.h>;#include <fcntl.h>;#include <sys/types.h>;#include <sys/stat.h>;#include <sys/wait.h>;#include <sys/sem.h>

void fibonacci (int nr){

int i;

for (i = 1; i <= nr; i++){

if (fork() == 0) {

if (i == 1 || i == 2) {

exit (1);

}

int a, b, c; a = 1, b = 1, c; int j;

for (j = 3; j <= i; j++){

c = a + b; a = b; b = c;

}exit (c);

}}

int status;

while (wait(&status) > 0) {

if (WIFEXITED(status) != 0) {

printf ("%d ", WEXITSTATUS(status));

}}

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

}

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

if (argc != 2) {

puts ("Numar invalid de argumente");

return -1;

}

int nr; // numarul primit ca argument

sscanf (argv[1], "%d", &nr);

fibonacci (nr);

return 0;

}

//fibonacci secvential 2

#include <stdlib.h>;#include <stdio.h>;#include <string.h>;#include <unistd.h>;#include <fcntl.h>;#include <ctype.h>;#include <sys/stat.h>;#include <sys/types.h>;#include <sys/wait.h>

void fibonacci (int n){

int firstVal = 1, secondVal = 1, aux;

for (int i = 0; i < 2 && n > 0; ++i, n--) {

printf("%d ", 1);

}

// IMPORTANT: trebuie fflush

fflush(stdout);

while (n != 0) {

if (fork() == 0) {

aux = firstVal + secondVal;

firstVal = secondVal;

secondVal = aux;

n--;

printf("%d ", aux);

fflush(stdout);

} else {

wait(NULL);

break;

} }}

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

if (argc != 2){

puts ("Numar invalid de argumente");

return -1;

}

int n; // numarul de ordine din sirul Fibo

sscanf (argv[1], "%d", &n);

fibonacci (n);

return 0;}