Task1

Global Μεταβλητές

```
ucontext_t co_main, co_read, co_write;
int counter_read, counter_write, terminating, fd_in, fd_out;
char *table;
```

Task1.c

Main

```
fd_in = open(file_in, O_RDONLY, S_IRWXU);
fd_out = open(file_out, O_RDWR|O_CREAT,
S_IRWXU);

dup2(fd_in, STDIN_FILENO);
dup2(fd_out, STDOUT_FILENO);

init(&co_main);
create(&co_write,pipe_write,writing,&co_read);
create(&co_read,pipe_read,reading,&co_main);
switchto(&co_main, &co_write);

diff_func();
close(fd);
mycoroutines_destroy();
```

Pipe read

```
While(1){
    while(counter_read!=counter_write){
        character = table[counter_read];
        counter_read++;
        write(STDOUT_FILENO, &character, 1);
     }
     if(terminating==1)  {pipe_close();  return}
      else{counter_read=0;
     mycoroutines_switchto(&co_read, &co_write);
      }
}
```

Pipe Write

```
while(1){
   while(counter_write != SIZE){
       scanf("%c", &character);
       if(check==EOF){terminating=1;return;}
       table[counter_write]=character;
       counter_write++;
   }
  mycoroutines_switchto(&co_write, &co_read);
   counter_write=0;
}
```

Mycontext.c

int mycoroutines_init(ucontext_t *main_con){

```
getcontext(main_con);
return(0);
```

int mycoroutines_create(ucontext_t *co, void func(),void* arg, ucontext_t *next){

```
getcontext(co);
co->uc_link=next;
co->uc_stack.ss_sp=(char*)malloc(STACK_SIZE);
co->uc_stack.ss_size=STACK_SIZE;
co->uc_stack.ss_flags=0;
makecontext(co,func,1,arg);
return(0);
```

int mycoroutines_switchto(ucontext_t *old_co, ucontext_t *new_co){

```
swapcontext(old_co, new_co);
return(0);
```

int mycoroutines_destroy(ucontext_t *co){

```
free(co->uc_stack.ss_sp);
co->uc_link=NULL;
co->uc_stack.ss_size=0;
return(0);
```

Diff_func()

```
lseek(fd_in, 0, SEEK_SET);
lseek(fd_out, 0, SEEK_SET);
do{
   read(fd_in, &cin, 1);
   read(fd_out, &cout, 1);
    if(cin != cout){
        fprintf(stderr,"not match\n");return;
    }
}while(check_in!=0 && check_out!=0);
if(check_in!=0 || check_out!=0){
   fprintf(stderr,"not match2\n");return;
}
```