

Контрольная работа №2.

1	2	3	4
+	+	FD	

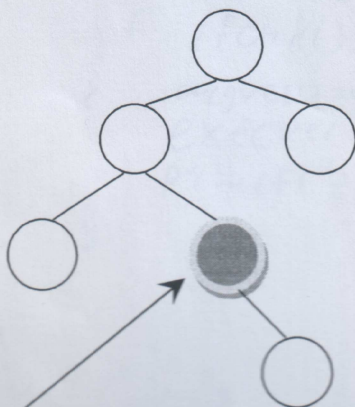
(Статья 1, УК ПРАК. За хранение и использование запрещённых материалов, а также списывание, вопрос или ответ соседу по теме или нет – поход на комиссию)

Вариант 1.

ФИО: Казыриг Макен и Игоревич

Задание 1. Написать программу.

После запуска программы, она должна породить следующее «генеалогическое» древо процессов:



- причём этот процесс должен выполнить код из выполняемого файла `"/usr/bin/virus"`.

Замечание: Последовательность рождения процессов на ваше усмотрение.

Задание 2. Написать программу.

На стандартный вход программе подаются имена двух различных текстовых файлов. Ваша программа должна уничтожить содержимое одного из файлов, и скопировать на его место содержимое другого файла.

Замечание: Использовать функции чтения/записи только низкого уровня.

Задание 3. Написать программу.

На stdin программе подаётся имя текстового файла. Строки указанного текстового файла, содержащие слово «Copyleft», должны выводиться на поток stderr. Все ОСТАЛЬНЫЕ строки вывести на поток stdout.

Замечания: Использовать функции чтения/записи только высокого уровня. Слово «copyleft» считать отличным от «Copyleft».

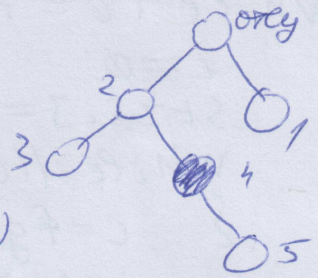
Задание 4. Написать программу.

На stdin программе подаются 10 двоичных целых чисел. Реализовать в программе две вспомогательные функции:

- 1) Возвращающую сумму двух чисел, переданных в функцию;
- 2) Возвращающую разность двух чисел, переданных в функцию.

Положить указатели на эти функции в массив `a_Operations[2]`. Подсчитать: «(сумма чётных чисел) минус (сумма нечётных чисел)» с помощью `a_Operations`. Замечание: Вызов вспомогательных функций напрямую запрещён.

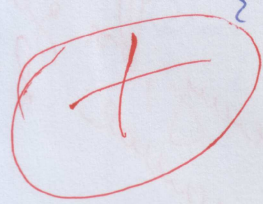
ΦΙΟ: Καζυριγ Μακεμυ Μορβιμ
№1



```

int main(int argc, char *argv[], char *env)
{
    if (fork() != 0) // ποχβιμια ①, οσραμια β ②;
    {
        if (fork() == 0) // β ②
        {
            if (fork() != 0) // ποχβιμια ③, οσραμια β ④
            {
                if (fork() == 0) // β ④
                {
                    if (fork() != 0) // ποχβιμια ⑤, οσραμια β ⑥
                    {
                        argv[0] = "virus";
                        execl("usr/bin/virus", argv, env);
                        printf("exes no cpaδotaλ");
                    }
                }
            }
        }
    }
}
return 0;

```

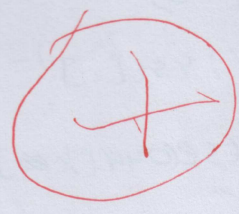


№2

```

int main(int argc, char *argv[])
{
    int fd1, fd2, i;
    char *buf;
    fd1 = open(argv[1], WR_ONLY, 0666);
    fd2 = open(argv[2], RD_ONLY, 0666);
    while (fread(fd2, buf, 100) == 100)
        write(fd1, buf, 100);
    write(fd1, buf, i);
    close(fd1);
    close(fd2);
    return 0;
}

```



№3

```

int main() ✓
{
    int i, k;
    char *f_name = NULL;
    File * myfile;
    char * name = NULL;
    char * str = NULL;
    ✓ char c; name = malloc(256);
    scanf("%s", name);
}

```

```

my_cmp(char *string)
{
    int i;
    char *copy = "copyleft";
    for (i=0; i<8; i++) ✓
        if (string[i] != copy[i])
            return 0;
    return 1;
}

```

Упомяне в "прымой" адрес sizeof использ


```
myFile = fopen(name, "r"); ! = -1 (and the EOF)
```

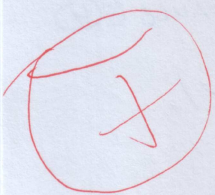
```
while (c = fgetc(myFile) != EOF)
```

```
{
    i = 0;
    str[i] = c;
    while ((c != '\n') && (c != EOF))
    {
        c = fgetc(myFile);
        i++;
        str[i] = c;
    }
    k = 0;
```

```
while (k < i)
    if (my_cmp(str[k], str[i]))
    {
        fprintf(stderr, "%s", str);
        break;
    }
```

```
else
    k++;
if (k >= i)
    fprintf(stdout, "%s", str);
```

```
}
return 0;
```



char: ?
no memory, for generation of my.c
so the cause of my.c

NO4

```
int sum2(char *first, char *second)
{
    int st = 1;
    int i, k;
    int rez = 0;
    i = 0;
    while (first[i] != '\0')
        i++;
    while (second[k] != '\0')
        k++;
    k--;
    while (i > 0)
    {
        rez = rez + first[i] * st;
        st *= 2;
        i--;
    }
    st = 1;
    while (k > 0)
    {
        rez += second[k] * st;
        st *= 2;
        k--;
    }
}
```

```
int rez(char *first, char *second)
{
    int st = 1;
    int i, k;
    int f = 0, s = 0;
    i = 0;
    k = 0;
    while (first[i] != '\0')
        i++;
    while (second[k] != '\0')
        k++;
    i--;
    k--;
    while (i > 0)
    {
        f += first[i] * st;
        st *= 2;
        i--;
    }
    st = 1;
    while (k > 0)
    {
        s += second[k] * st;
        st *= 2;
        k--;
    }
    return (f - s);
}
```


no 3

```

int main()
{
  int i;
  char *f_name = NULL;
  File * myFile;
  char *name;
  scanf("%s", name);
}

```

```

my_cmp(char *first)
{
  int i;
  char *copy = "copy left";
  for(i=0, i<8, i++)
  {
    if (first[i] != copy[i])
      return 0;
  }
  return 1;
}

```

```

i=0;
myFile = fopen(name, "r");
while (c = fgetc(myFile) != EOF)
{
  i = 0;
  strcpy(s, "");
  while (c != '\n' && c != EOF)
  {
    c = fgetc(myFile);
    i++;
    strcpy(s, s + c);
  }
  while (k < i - 4)
  {
    if (my_cmp(s + k))
      fprintf(stderr, "%s", s + k);
    else
      k++;
  }
  if (k >= i - 4)
    fprintf(stdout, "%s", s + k);
}

```

1101
11170100
011100
~~1110~~ 000

no 4

```

char * sum2(char * first, char * second)
{
  int i, k;
  int offset;
  i = 0, k = 0;
  while (first[i] != '\0')
    i++;
  while (second[k] != '\0')
    k++;
  i = i - k;
}

```


εργασίες

```

if (i > k)
  s = i;
else s = k;
s++;
while ((i > 0) && (k > 0))

```

$max = \max(\text{char } x) \text{ min}(\text{char } s + \text{sizeof}(\text{char}))$
 $OST = \emptyset$
ΦΙΟ: Καζβερης Μ. Λ

```

s[first] = (first[i] + second[k] + OST) ^ 2;
OST = (first[i] + second[k] + OST) ^ 1;
i--;
k--;
s--;

```

} ~~while~~

```

while (i > 0)
  s[first] = (first[i] + OST) ^ 2;
  OST = (first[i] + OST) ^ 1;
  i--;

```

while (k > 0)

```

  s[s] = (second[k] + OST) ^ 2;
  OST = (second[k] + OST) ^ 1;
  k--;
  s--;

```

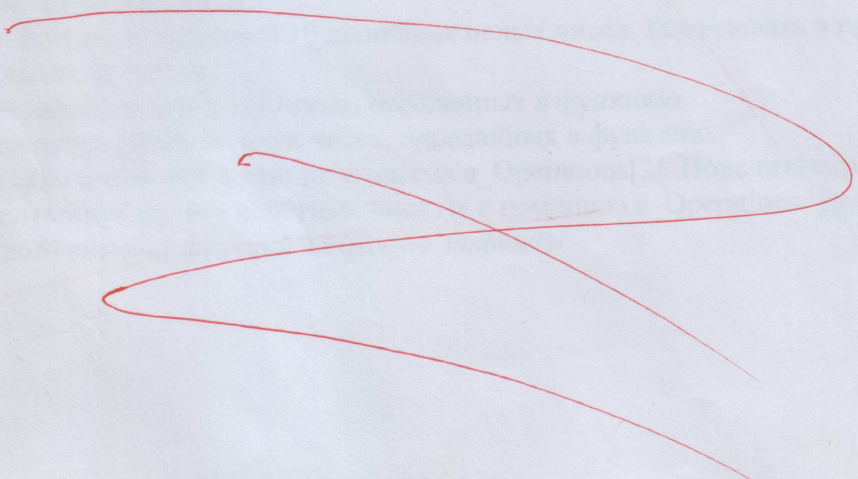
}

OST = OST;

return OST;

}

char * max(char *first, char *second)




```
int sum? (char * f; int, char * second)
```

```
int st;
```

```
int i, k;
```

```
while
```

Задача не решена.
Она требует больше
времени.