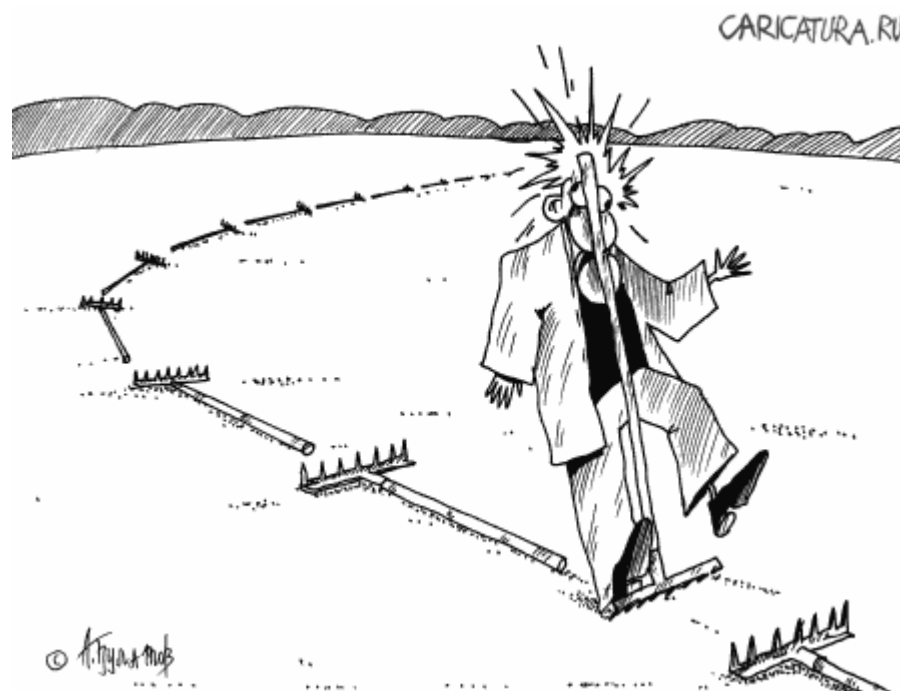
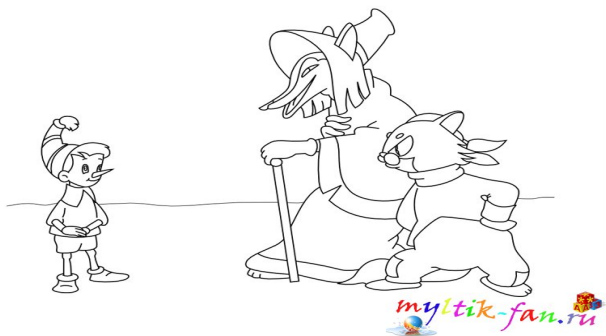


# Грабли в С



# Переменные

- Требуется инициализация
- На Ejudge мусор в переменных будет



# Типы данных

- char имеет размер 1 байт и
- sizeof(char) ≤ sizeof(short) ≤ sizeof(int) ≤ sizeof(long) ≤ sizeof(long long)
- А сколько конкретно....
- sizeof() их знает!

# Ветвления?

- `if(a=b)`  
    {  
    }

# Ветвления!

- `if(7==a)`  
    `{`  
    `}`

# switch

- switch(c)  
  case 1: a++;  
  case 2: b++;

# Порядок действий

```
int i = 5;
```

```
i = ++i + ++i;
```

# ЦИКЛЫ

```
int i,j;
```

```
for(j=0,i=0;i<n;i++);
```

```
j+=i*i;
```



Адрес=int

```
scanf(«%d», n);
```

# Заграница нам поможет?

```
int a[n];
```

```
for(i=0;i<=n;i++) a[i]=i;
```



- В наше время верить  
нельзя никому.....

# А чему можно?

- Стандарт
- Предупреждения компилятора
- Assert
- Отладочная печать
- Инструментальные средства

# Приоритет операций

- ++ префиксный    () [] .    →
- - ! ~ (type) \* & sizeof()
- \* / %
- + -
- « »
- < <= >=
- == !=
- &
- ^
- |
- &&
- ||
- ?:
- = += -= \*= /= %= <<= >>= &= ^= |=
- ++ суффиксный

# Предупреждения компилятора

gcc

-Wall

-Werror=format

-Werror=return-type

-Werror=vla

-Werror=uninitialized

-Werror=maybe-uninitialized

-Werror-implicit-function-declaration

# assert

```
if(num_of_opened_brackets < 0)
{
    printf( «NO\n»);
    return 1;
}
```

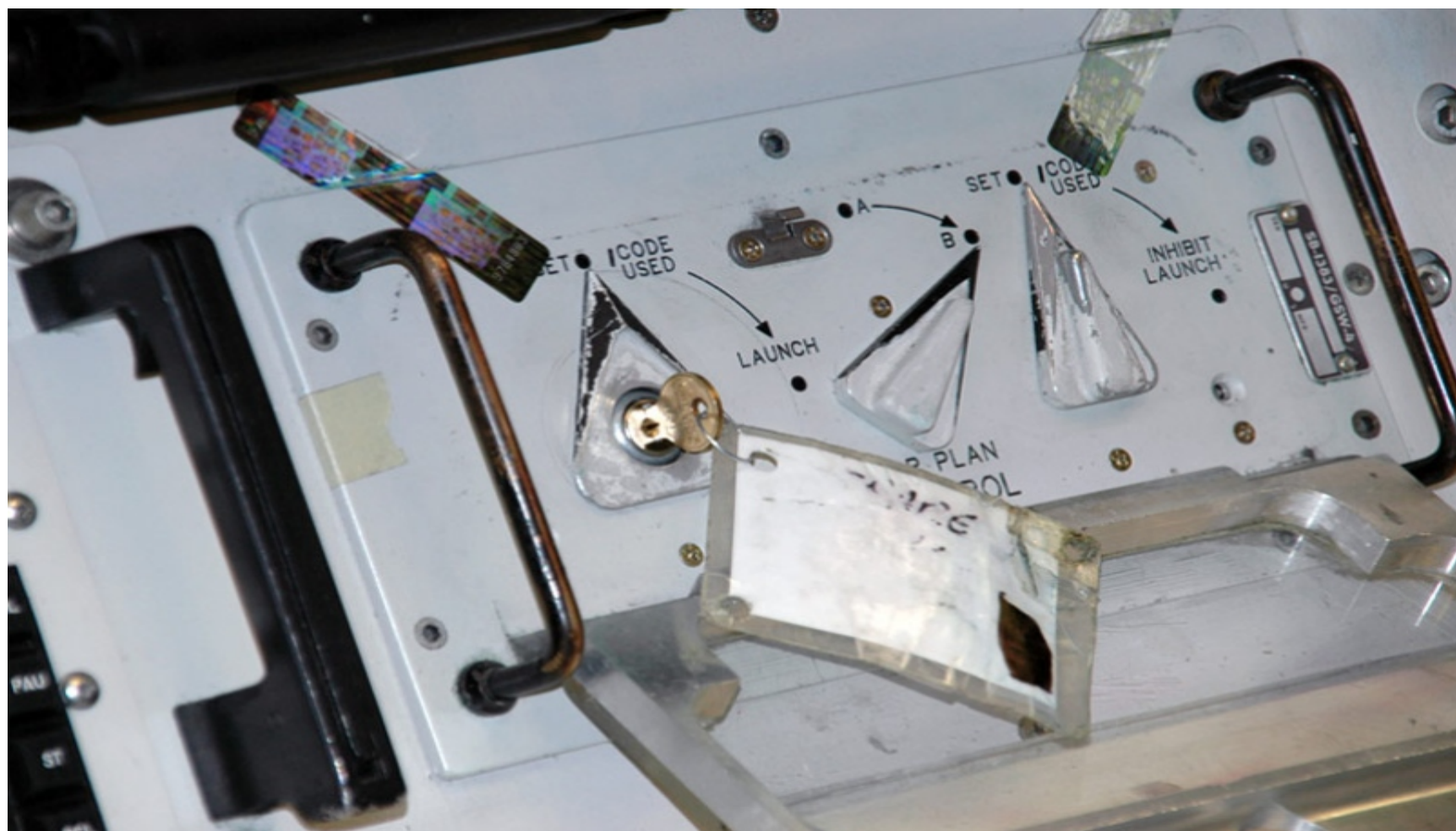
# Ни одна программа без assert'a работать не может



- Фраза приписывается заведующему кафедрой теоретической и прикладной информатики Александру Геннадьевичу Тормасову



# Проверяйте входные данные!



# Valgrind

Инструмент для

- отладки использования памяти
- обнаружения утечек памяти
- профилирования

```
#include <stdio.h>
int main()
{int a,b,sum;
sum=a+b;
scanf("%d",&a);
scanf("%d",&b);
printf("%d\n",sum);
}
```

# Компиляция с отладочной информацией

```
gcc -g sum.c
```

```
valgrind ./a.out
```

```
==10358== Memcheck, a memory error detector
==10358== Copyright (C) 2002-2012, and GNU GPL'd, by Julian
Seward et al.
==10358== Using Valgrind-3.8.1 and LibVEX; rerun with -h for
copyright info
==10358== Command: ./a.out
==10358==
3 4
==10358== Conditional jump or move depends on uninitialised value(s)
==10358==    at 0x374D849F96: vfprintf (in /usr/lib64/libc-2.15.so)
==10358==    by 0x374D850878: printf (in /usr/lib64/libc-2.15.so)
==10358==    by 0x40059E: main (sum.c:7)
==10358== Uninitialised value was created by a stack allocation
==10358==    at 0x40054C: main (sum.c:3)
```

# Динамическое выделение памяти

```
#include <stdlib.h>
void *x;
int main()
{
x=malloc(12);
}
```

- `gcc -g malloc.c`
- `[dmitry@judge vg]$ valgrind ./a.out`
- `==10039==` Memcheck, a memory error detector
- `==10039==` Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
- `==10039==` Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright info
- `==10039==` Command: `./a.out`
- `==10039==`
- `==10039==`
- `==10039==` HEAP SUMMARY:
- `==10039==`   in use at exit: 12 bytes in 1 blocks
- `==10039==`   total heap usage: 1 allocs, 0 frees, 12 bytes allocated
-