

Задание 1. Сколько байт занимает значение типа `double`? А сколько — `(double*)`? Сколько `(double&)`?

Задание 2. Что напечатает программа?

```
1 int acc = 2;
2 int foo(int i){
3     return acc++;
4 }
5
6 int main(void)
7 {
8     printf("%d, %d, %d", foo(1), foo(2), foo(3));
9     printf("%d, d, d", (foo(4), foo(5), foo(6)));
10    return 0;
11 }
```

Задание 3. Перед вами фрагмент кода на C++. Какое значение он напечатает? В чем состоит возможная ошибка?

```
1 double d = 1000000*100000*100000*0.39487;
2 printf("%f", d);
```

Задание 4. Перед вами фрагмент кода на C++. Что должен делать данный код по замыслу программиста? В чем состоит возможная ошибка? Как ее исправить?

```
1 const char* password = "TopSecretP@zzword";
2 char sz[100];
3 cin.getline(sz, 100);
4 if(sz == password){
5     printf("Access granted!\n");
6 }else{
7     printf("Access denied!\n");
8 }
```

Задание 5. Перед вами функция на языке C++. Что по задумке автора должна делать эта функция? Что она делает на самом деле? Как ее исправить?

```
1 void swap(int a, int b){
2     int tmp = a;
3     a = b;
4     b = tmp;
5 }
```

Задание 6. Перед вами функция на языке C++. Что она вычисляет? Почему эта функция реализована неоптимально? Как переписать ее оптимально?

```
1 int foo(int n){
2     switch(n){
3         case 0: return 0;
4         case 1: return 1;
5         default: return foo(n-1) + foo(n-2);
6     }
7 }
```

Задание 7. Какую ошибку допустил программист при работе с памятью?

```
1 int *p = new int[10];
2 for(int i = 0; i <= 10; ++i)
3     p[i] = i*i;
4 delete [] p;
```

