

На всякъде, където изберете верен отговор *няма да се компилира*, дайте кратка обосновка.

Задача 1: Какво ще се изведе на конзолата след изпълнението на задачата:

```
#include <iostream>

const int MAX_NAME_SIZE = 64;

class Person {
    int age;
    char name[MAX_NAME_SIZE + 1];
};

int main() {
    Person p;
    std::cout << p.age << std::endl;
}
```

- a) 0 б) undefined в) няма да се компилира

Задача 2: Какво ще се изведе на конзолата след изпълнението на задачата:

```
#include <iostream>

const int MAX_NAME_SIZE = 64;

class Person {
    int age;
    char name[MAX_NAME_SIZE + 1];
public:
    int getAge() const {
        return age;
    }
};

int main() {
    Person p;
    std::cout << p.getAge() << std::endl;
}
```

- a) 0 б) undefined в) няма да се компилира

Задача 3: Какво ще се изведе на конзолата след изпълнението на задачата:

```
#include <iostream>

const int MAX_NAME_SIZE = 64;

class Person {
    int age;
    char name[MAX_NAME_SIZE + 1];
public:
    Person(const char* name, int age) {
        strcpy(this->name, name);
        this->age = age;
    }
}
```

```

        int getAge() const {
            return age;
        }
};
int main() {
    Person p;
    std::cout << p.getAge() << std::endl;
}

```

- а) 0 б) undefined в) няма да се компилира

Задача 4: Какво ще се изведе на конзолата след изпълнението:

```

#include <iostream>

class A {
    int a;
public:
    A(int a) {
        this->a = a;
    }
    void setValue(int number){
        this->a = number;
    }
};
class B {
    double b;
    A a;
public:
    B(double b,int number) {
        this->b = b;
        a.setValue(number);
    }
    double getDoubleValue() const {
        return b;
    }
};

int main() {
    B b (8.5,4);
    std::cout << b.getB();
}

```

- а) 8.5 б) undefined в) няма да се компилира

Задача 5:

В какъв ред ще бъдат инициализирани член-данните за дадения клас и конструктор:

```

#include <iostream>

const int MAX_NAME_LENGTH = 64;

class Person {

```

```

    char gender;
    int fn;
    int age;
    double averageGrade;
public:
    Person(char gender, int fn, int age, double averageGrade) :
        fn(fn), averageGrade(averageGrade), gender(gender), age(age) {}
};

```

Задача 7: Какво ще се отпечата на конзолата?

```

class Test {
    void g() const {
        t();
    }
public:
    void t() {
        std::cout << "Hello" << std::endl;
    }
    void f() const {
        g();
    }
};

int main() {
    Test t;
    t.f();
}

```

а) Hello б) undefined в) няма да се компилира

Задача 8: Какво ще се изведе на екрана след изпълнение

```

#include <iostream>

class A {
    int a;
public:
    A() {
        std::cout << "Default constructor of A\n";
    }
    A(int a) : a(a) {}
    ~A() {}
}

```

```

        std::cout << "Destructor of A\n";
    }
};

class B {
    int b;
    A a;
public:
    B() {
        std::cout << "Default constructor of B\n";
    }
    B(int b) : b(b) {}
    ~B() {
        std::cout << "Destructor of B\n";
    }
};

class C {
    A a;
    B b;
public:
    C() : b(), a() {
    }
    ~C() {
        std::cout << "Destructor of C\n";
    }
};

int main() {
    C arr1[2];
    C* arr2 = new C[2];
    return 0;
}

```

Задача 9: Напишете клас, представляващ човек. Всеки човек трябва да има име с максимална дължина 64 символа и възраст. Напишете подходящи конструктори, селектори и мутатори.