

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"



ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

### ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС "БАКАЛАВЪР ПО КОМПЮТЪРНИ НАУКИ"

### ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листове.
- Пишете само на предоставените листове, без да ги разкопчавате.
- Решението на една задача трябва да бъде на същия лист, на който е и нейното условие (т.е. може да пишете отпред и отзад на листа със задачата, но не и на лист на друга задача).
- Ако имате нужда от допълнителен лист, можете да поискате от квесторите.
- На един лист не може да има едновременно и чернова, и белова.
- Черновите трябва да се маркират, като най-отгоре на листа напишете "ЧЕРНОВА".
- Ако решението на една задача не се побира на нейния лист, трябва да поискате нов бял лист от квесторите. Той трябва да се защити с телбод към листа със задачата.
- Всеки от допълнителните листове (белова или чернова) трябва да се надпише най-отгоре с вашия факултетен номер.
- Черновите също се предават и се защитават в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Handwritten notes and code snippets:

```

int x & r = arr;
int x & r = P;
r++;
int x r2 = P;

q = P;
int x q = P-2;
int x q = &--P;
int x q = &P;

arr[i] => *(arr+i)
&arr[0] => &(arr[0])

&3
int & x = y;

int x p = arr+1;
P = arr+2;

```

~~arr[0] => \*(arr+0) => \*arr~~

Задача 1. Задачата да се реши на езика C++.

1) Нека е дефиниран масивът

```
int arr[] = { 1, 2, 3 };
```

Срещу всеки от изразите да се посочи каква ще бъде неговата оценка.

```
arr[1] == *(arr+2) false
arr == &arr[0] true
(arr+1) == &arr[1] true
*arr == arr[0] true
```

~~int\*\* p = &v;~~ int \* p = arr+2;

2) Нека е дадена следната дефиниция:

```
void mystery(const char* str)
{
    while (*str && *(str+1)) {
        std::cout << *str;
        str += 2;
    }
}
```

Да се посочи какво ще изведе на екрана обръщението:

```
mystery("abcdef");
```

fedcba

3) Нека са дадени следните дефиниции:

```
char s1[] = "Hello";
char s2[] = "world!";
char result[80];
```

Да се довърши програмният фрагмент, така че след изпълнението му в result да се съхрани коректното представяне на низа "Hello world!". На празните места трябва да се попълнят имената на подходящи стандартни функции за работа с низове.

```
_____ (result, s1);
_____ (result, " ");
_____ (result, s2);
```

```
int arr[] = { 1, 2, 3, 4, 0 };
*(arr + 2) *= 10; // arr[2] = arr[2] * 10;
for (int* p = arr; *p; p++)
    std::cout << *p << " ";
```

4) Да се довърши кодът на рекурсивните функции, така че f да проверява дали символният низ, сочен от word, се съдържа като подниз в text. За определеност считаме, че празният низ се съдържа във всеки друг.

```
bool g(const char* text, const char* word)
{
    if (!word) return true;
    if (!*text) return false;
    if (*word != *text) return false;
    return g(text+1, word+1);
}
```

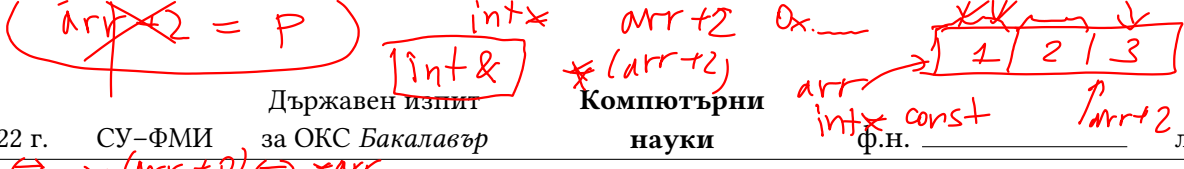
```
bool f(const char* text, const char* word)
{
    if (!*word) return true;
    if (!*text) return false;
    return g(text, word) ||
           f(text+1, word);
}
```

5) Да се посочи какво ще изведе на екрана даденият по-долу фрагмент:

```
char arr[3][3] = { 'a', 'b', 'c',
                  'd', 'e', 'f',
                  'g', 'h', 'i' };
for (int i = 0; i < 3; ++i)
    std::cout << arr[2-i][i];
```

6) Да се посочи какво ще изведе на екрана даденият по-долу фрагмент:

```
double var = 5 / 2;
std::cout << var;
```



~~arr = p~~ null ptr

**Задача 2.** Задачата да се реши на езика C++.

1) Освен конструктора по подразбиране (default constructor), кои други функции влизат в “голямата четворка” (функциите от т.нар. “rule-of-3”)? Да се попълнят имената им в полетата долу:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

2) Нека е дадена дефиницията:

```
class foo {  
public:  
    virtual void f() {};  
    void g() {};  
};
```

Срещу всеки от редовете, които извикват f или g, да се запише “статично” или “динамично” според вида свързване, който ще се използва за тях.

```
foo obj;  
foo& ref = obj;  
obj.f(); _____  
obj.g(); _____  
ref.f(); _____  
ref.g(); _____
```

3) Нека са дадени следните дефиниции:

```
class base {  
public: int a;  
private: int b;  
};  
class derived : protected base { };
```

Да се посочи каква ще бъде видимостта на променливите a и b в класа derived – public, protected или private.

- Видимост на a: \_\_\_\_\_
- Видимост на b: \_\_\_\_\_

4) Нека класът X е абстрактен. Срещу всяко от твърденията да се посочи “да” или “не” според това дали е вярно:

- Могат да се създават обекти от тип X: \_\_\_\_\_
- Могат да се създават референции (reference) към обекти от тип X: \_\_\_\_\_

5) Нека е дадена следната дефиниция:

```
struct s {  
public:  
    static int var;  
    s() { var = 5; }  
};  
int s::var = 0;
```

Да се посочи какво ще изведе следният фрагмент:

```
std::cout << '(' << s::var << ')';  
s obj1;  
obj1.var = 10;  
s obj2;  
std::cout << '-' << s::var << '-';
```

6) Да се допълни дефиницията на класа test, така че функцията f да бъде чиста виртуална (pure-virtual) и класът да може коректно да се използва като основа на полиморфна йерархия.

```
class test {  
public:  
    _____ void f() _____;  
};
```

7) Да се допълни дефиницията на шаблона Array, така че функцията test да се компилира коректно и да извежда на стандартния изход 55.

```
template < typename T >  
class Array {  
    static const size_t size = 10;  
    T data[size];  
public:  
    T& _____ at(size_t index) {  
        if (index >= size || index < 0)  
            throw std::out_of_range("error");  
        return data[index];  
    }  
};  
  
void test() {  
    Array<int> a;  
    a.at(0) = 5;  
    std::cout << a.at(0);  
    Array<Array<int>> b;  
    b.at(0) = a;  
    std::cout << b.at(0).at(0);  
}
```