

1. Сортировка показана в фрагменте кода
- ```
void Sort(vector<int>& values) {
 std::make_(values.begin(), values.end());
 for (auto i = values.end(); i != values.begin(); --i) {
 std::pop_(values.begin(), i);
 }
}
```
- A) Сортировка вставками  
B) Пирамидальная сортировка  
C) Пузырьковая сортировка  
D) Сортировка слиянием  
E) Быстрая сортировка
2. Сигнал, обнаружение недействительного образа функции, такого, как запрещенная инструкция
- A) SIGFPE  
B) SIGILL  
C) SIGABRT  
D) SIGINT  
E) SIGTERM
3. #include <iostream>  
using namespace std;  
int main()  
{  
 int ans = 27;  
 ans -= 7;  
 cout << ans << ", ";  
 ans += 10;  
 cout << ans << ", ";  
 ans \*= 2;  
 cout << ans << ", ";  
 ans /= 3;  
 cout << ans << endl;  
 return 0;  
}
- Результат работы программы будет следующим:
- A) 20, 30, 60, 20  
B) 37, 20, 60, 20  
C) 20, 30, 60, 37  
D) 20, 30, 60, 0  
E) 37, 30, 60, 20

```

4. #include <iostream>
 #include <algorithm>
 using namespace std;
 int source[] = { 11, 44, 33, 11, 22, 33, 11, 22, 44 };
 int pattern[] = { 11, 22, 33 };
 int main()
 {
 int* ptr;
 ptr = search(source, source + 9, pattern, pattern + 3);
 if(ptr == source + 9)
 cout << "Совпадения не найдено\n";
 else
 cout << "Совпадение в позиции " << (ptr - source) << endl;
 return 0;
 }

```

Программа выводит на экран результат:

- A) Совпадение в позиции 4
  - B) Совпадение в позиции 1
  - C) Совпадение в позиции 0
  - D) Совпадение в позиции 2
  - E) Совпадение в позиции 3
5. Пирамидальная сортировка выполняется в среднем за время
- A)  $\Omega(n)$
  - B)  $O(n \log n)$
  - C)  $O(n^2)$
  - D)  $\Omega(n^2)$
  - E)  $O(n \log n^2)$
6. Пузырьковая сортировка выполняется в среднем за время
- A)  $O(n \log n^2)$
  - B)  $O(n^2)$
  - C)  $\Omega(n^2)$
  - D)  $\Omega(n)$
  - E)  $O(n \log n)$
7. Заголовочный файл, содержащий прототипы для функций, выполняющие ввод из строк в память
- A) <cassert>
  - B) <sstream>
  - C) <cstring>
  - D) <memory>
  - E) <fstream>

8. Операция <<

- A) операция поместить в поток
- B) логическая операция
- C) операция присвоения
- D) операция взять из потока
- E) операция сравнения

9. Результат работы программы

```
#include<iostream>
using namespace std;
int main()
{int i=0,k=0;
int arrsize = 14;
int a[14] = {4, 5, 3, 2, 1, 9, 1, 4, 5, 3, 2, 7, 9, 1};
for(int i = 0; i<arrsize; i++){
if (i %2 ==0)
k++;}
cout<<k;}
```

- A) 7
- B) 9
- C) 14
- D) 0
- E) 2

10. Функция вычисления суммы целых чисел, находящихся между двумя заданными (начальное и конечное включительно), имеет прототип `int sum(int start, int end)`, определение функции, с использованием рекурсии выглядит следующим образом

- A) `{if (end == start) return 0; else return end + sum(start, end -1);}`
- B) `{if (end == start) return 2; else return end + sum(start, end -1);}`
- C) `{if (end == start) return 1; else return end + sum(start, end -1);}`
- D) `{if (end == start) return 1; else return end + sum(start, end -2);}`
- E) `{if (end == start) return start; else return end + sum(start, end -1);}`