Q-20

Due No due date **Points** 15 **Questions** 17 **Time Limit** None **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	21 minutes	12 out of 15

Submitted Jul 21 at 9:32pm

Question 1

2 / 2 pts

Match each item with the correct definition below.

Correct! **Elements always allocated on the heap** — vector

Correct! **const int *array** — Elements may not be modified;

Correct! **sizeof(a) / sizeof(a[0])** — Elements in array using arithme

Correct! **x = 0; for (auto e : a) x += e;** — Cumulative algorithm

Question 2

1 / 1 pts

What is printed here? (Assume all includes have been added.)

```
int odds(int a[], size_t len)
{
    int sum = 0;
    for (size_t i = 0; i < len; i++)
        if (a[i] % 2 == 1) sum += a[i]++;
    return sum;
}

int main()
{
    int a[] = {1, 3, 5};
    cout << odds(a, 3) << odds(a, 2)
        << odds(a, 1) << endl;

}
```

- 999
- 941
- Correct! 900
- 300
- Does not compile

Question 3

1 / 1 pts

What is the correct prototype for mystery? (It may modify the array.)

```
const int a[] = {2, 4, 6, 8};
cout << mystery(a, 4) << endl;
```

- int mystery(int a, size_t n);
- Correct! int mystery(int *a, size_t n);
- void mystery(const int a[], size_t n);
- int mystery(int[] a, size_t n);
- int mystery(int a*, size_t n);

Question 4

1 / 1 pts

What is printed here? (Assume all includes have been added. Assume 4-bytes per int, 8 bytes per pointer.)

```
int main()
{
    int a[] = {2, 4, 6, 8};
    cout << sizeof(a) / sizeof(a[0]) << endl;
}
```

Correct!

- (•) 4
- ( ) 2
- ( ) 1
- ( ) Does not compile

Question 5

1 / 1 pts

What is printed here? (Assume all includes have been added. Assume 4-bytes per int, 8 bytes per pointer.)

```
size_t len(const int* a, const int* b)
{
    return b - a;
}

int main()
{
    int a[] = {2, 4, 6, 8};
    cout << len(begin(a), end(a)) << endl;
}
```

Correct!

- (•) 4
- ( ) 2
- ( ) 1
- ( ) Does not compile

Question 6

1 / 1 pts

What is printed here? (Assume all includes have been added. Assume 4-bytes per int, 8 bytes per pointer.)

```
size_t len(const int* a, const int* b)
{
    return b - a;
}

int main()
{
    int a[] = {2, 4, 6, 8};
    cout << len(a, a + 3) << endl;
}
```

- ( ) 4

Correct!

- (•) 3
- ( ) 2
- ( ) Does not compile

Question 7

0.5 / 0.5 pts

An array passed to a function is passed by reference.

- ( ) True

Correct!

- (•) False

Question 8

0.5 / 0.5 pts

If ***p*** points to the first element in ***[1, 3, 5]*** then ***cout << *p++*** prints ***1***.

Correct!

- (•) True
- ( ) False

Question 9

0 / 0.5 pts

The expression ****p++*** means the same as ****(p++)***.

Correct Answer

- ( ) True

You Answered

- (•) False

Question 10

0.5 / 0.5 pts

If ***p*** points to the first element in ***[1, 3, 5]*** then ***cout << ++*p*** prints ***2***.

Correct!

- (●) True
- ( ) False

Question 11

0 / 0.5 pts

Before passing an array to a function, ***sizeof(a)*** will tell you the array's allocated size, but not the number of elements.

Correct Answer

- ( ) True

You Answered

- (●) False

Question 12

0.5 / 0.5 pts

For embedded systems, arrays are preferred over ***vector***.

Correct!

- (●) True
- ( ) False

Question 13

0 / 1 pts

What does this function do?

```
int mystery(const int a[], size_t n)
{
    int x = n - 1;
    while (n > 0)
    {
        n--;
        if (a[n] > a[x]) x = n;
    }
    return x;
}
```

- ( ) Does not compile
- ( ) Returns the largest number in the array

You Answered

- (●) Returns the index of the first occurrence of the largest number in the array

Correct Answer

- ( ) Returns the index of the last occurrence of the largest number in the array
- ( ) Returns the smallest number in the array

Question 14

1 / 1 pts

Below is a cumulative algorithm using an array and an iterator-based loop. What is printed? (Assume all includes have been added, etc.)

```
double average(const int *beg, const int *end)
{
    double sum = 0;
    size_t count = end - beg;
    while (beg != end) sum += *beg++;
    return sum / count;
}

int main()
{
    int a[] = {2, 4, 6, 8};
    cout << average(begin(a), end(a)) << endl;
}
```

Correct!

- (●) 5
- ( ) 4
- ( ) 6
- ( ) Endless loop when run; likely crashes.
- ( ) Does not compile

Question 15

1 / 1 pts

What is printed?

```
int mystery(const int a[], size_t n)
{
    int x = 0;
    for (size_t i = 0; i < n; i++)
        if (a[i] < a[x]) x = i;

    return x;
}

int main()
{
    int a[] = {4, 2, 5, 2, 5, 4};
    cout << mystery(a, 6) << endl;
}
```

- 0
- Correct! 1
- 2
- 3
- None of these

Question 16

1 / 1 pts

Below is a cumulative algorithm using an array and a range-based loop. What is printed? (Assume this is inside *main()* with all includes, etc.)

```
int a[] = {2, 4, 6, 8};
int sum = 0;
for (auto e : a) sum =+ e;
cout << "sum->" << sum << endl;
```

- sum->20
- Does not compile. Cannot use range-loop on arrays.
- Correct! sum->8
- Does not compile; e is undefined.
- Compiles and runs, but results are undefined.

Question 17

0 / 1 pts

What does this function do?

```
int mystery(const int a[], size_t n)
{
    int x = a[n - 1];
    while (n > 0)
    {
        n--;
        if (a[n] > a[x]) x = a[n];
    }
    return x;
}
```

- Does not compile
- Correct Answer Returns the largest number in the array
- Returns the index of the first occurrence of the smallest number in the array
- Returns the index of the last occurrence of the smallest number in the array
- You Answered Returns the smallest number in the array