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
KEPT	Attempt 2	14 minutes	13.5 out of 15
LATEST	Attempt 2	14 minutes	13.5 out of 15
	Attempt 1	21 minutes	12 out of 15

Submitted Jul 21 at 9:48pm

Question 1

2 / 2 pts

Match each item with the correct definition below.

Correct! **How arrays are passed to functions** — by address

Correct! **int * const array** — Elements in may be modified; p

Correct! **x = a[0]; for (auto e: a) if (e > x) x = e;** — Extreme values algorithm

Correct! **auto p = a; while (p != end(a)) p++;** — Iterator-based loop

Question 2

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[])
{
    return sizeof(a) / sizeof(a[0]);
}

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

- 4
- Correct! 2
- 1
- 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.)

```
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
- 3 — Correct!
- 2
- Does not compile

Question 5

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
- 900 — Correct!
- 300
- Does not compile

Question 6

1 / 1 pts

What is the correct prototype for mystery? (It is not supposed to modify the array.)

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

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

Question 7

0 / 0.5 pts

After passing an array to a function, ***`sizeof(a)/sizeof(a[0])`*** will tell the number of elements in the array.

- True — You Answered
- False — Correct Answer

Question 8

0.5 / 0.5 pts

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

- True — Correct!
- False

Question 9

0.5 / 0.5 pts

For systems programming (such as operating systems), ***vector***s are used more often than arrays.

- ○ True
- Correct! ◉ False

Question 10

0.5 / 0.5 pts

An array passed to a function ***f(int *a, …)*** may have its elements changed.

- Correct! ◉ True
- ○ False

Question 11

0.5 / 0.5 pts

The elements of a ***vector*** are allocated on the heap.

- Correct! ◉ True
- ○ False

Question 12

0.5 / 0.5 pts

An array passed to a function decays to a pointer.

- Correct! ◉ True
- ○ False

Question 13

1 / 1 pts

What does this function do?

```
double mystery(const double a[], size_t len)
{
    double x = a[0];
    for (size_t i = 1; i < len; i++)
        if (a[i] > x) x = a[i];
    return x;
}
```

- ○ Does not compile
- Correct! ◉ Returns the largest number in the array
- ○ Returns the smallest number in the array
- ○ Undefined. Depends on the input.

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(a, a + 1) << endl;
}
```

- Correct! ◉ 2
- ○ 3
- ○ 4
- ○ 5
- ○ Does not compile

Question 15

0 / 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(a + 1, a + 3) << endl;
}
```

You Answered

- (•) 2
- ( ) 3
- ( ) 4
- ( ) 5
- ( ) Does not compile

Correct Answer

Question 16

1 / 1 pts

What is printed?

```
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;
}

int main()
{
    int a[] = {1, 2, 5, 2, 5, 4};
    cout << mystery(a, 6) << endl;
}
```

- ( ) 1
- ( ) 2
- ( ) 3
- ( ) 4

Correct!

- (•) None of these

Question 17

1 / 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
- ( ) Returns the index of the first occurrence of the smallest number in the array

Correct!

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