

Instructions: This homework assignment focuses on the basics of C++ arrays. The answers to the following questions can be determined from Chapters 3 through 8 of the lecture notes and Chapters 11 and 12 of the text. Assume any `#include` directives, variable declarations, etc, which are needed to make the given code syntactically correct.

Print your name and code your ID number correctly on the opscan form.

Mark Group 1 if you are in the 8TTh section and Group 2 if you are in the 10MWF section.

Turn in your completed opscan at class on April 7th or 8th, or to the 1044 GTAs in McB 116/118 between 12:00 and 3:00 on Thursday April 8th. Opscans will not be accepted late or at any other time.

1) Given the declaration: `float alpha[75];`

the logically valid range of index values for alpha is:

- | | |
|-----------------|------------------|
| 1) 0 through 75 | 2) 0 through 74 |
| 3) 1 through 75 | 4) 1 through 74 |
| 5) 1 through 76 | 6) none of these |

2) What is the output of the following program fragment?

```
int gamma[3] = {5, 10, 15};  
  
for (int i = 0; i < 3; i++)  
    cout << gamma[i] << ' ';
```

- | | |
|--|------------------|
| 1) 5 10 15 | 2) 5 10 |
| 3) 0 1 2 | 4) 0 1 |
| 5) It cannot be answered from the information given. | 6) none of these |

3) Given the array declaration: `int status[10];`

which of the following loops correctly stores zeros at each valid index of the array status?

- 1)

```
for (int i = 0; i <= 10; i++)  
    status[i] = 0;
```
- 2)

```
for (int i = 1; i < 10; i++)  
    status[i] = 0;
```
- 3)

```
for (int i = 1; i <= 10; i++)  
    status[i] = 0;
```
- 4)

```
for (int i = 0; i < 10; i++)  
    status[i] = 0;
```
- 5)

```
for (int i = 1; i <= 11; i++)  
    status[i] = 0;
```
- 6) none of these

For questions 4 and 5, consider execution of the code fragment:

```
int arr[5];
int i;

for (i = 0; i < 5; i++) {
    arr[i] = i + 2;
    if (i >= 3)
        arr[i-1] = arr[i] + 3;
}
```

4) What value will be contained in arr[1]?

- 1) 2 2) 3 3) 7 4) 8
5) none of these

5) What value will be contained in arr[3]?

- 1) 2 2) 3 3) 7 4) 8
5) none of these
-

6) What is the output of the following program fragment?

```
int alpha[5] = {100, 200, 300, 400, 500};
int i;

for (i = 4; i > 0; i--)
    cout << alpha[i] << ' ';
```

- 1) 400 300 200 100 2) 500 400 300 200 100
3) 500 400 300 200 4) 4 3 2 1
5) It cannot be answered from the information given. 6) none of these

7) Given a 5000-element integer array beta, which of the code fragments below could be used to print out the values of beta[0], beta[2], beta[4], and so forth?

- 1) for (int i = 0; i < 5000; i = i + 2)
 cout << beta[i] << endl;
- 2) for (int i = 0; i < 2500; i++)
 cout << beta[2*i] << endl;
- 3) for (int i = 0; i < 2500; i++)
 cout << beta[i]*2 << endl;
- 4) all of the above
5) 1 and 2 above
6) none of these

Questions 8 through 10 refer to the following incomplete program:

```
void FillEm(_____ arr1[], _____ arr2[], int length); // line 1
void Copy(_____ arr1[], _____ arr2[], int length);    // line 2

void main() {
    const int dim = 200;
    char alpha[dim];
    char beta[dim];

    FillEm(alpha, beta, dim); // Initialize arrays alpha and beta
    Copy(alpha, beta, dim);   // Copy all components of beta into alpha
}

void Copy(_____ arr1[], _____ arr2[], int length) {
    for (int Idx = 0; Idx < length; Idx++) {
        arr1[Idx] = arr2[Idx];
    }
    return;
}

void FillEm(_____ arr1[], _____ arr2[], int length) {
    // some initialization code goes here
}
```

- 8) What is the most appropriate way to fill the blank preceding the first formal parameter of FillEm () in line 1?
- | | |
|-------------------------|------------------|
| 1) int& | 2) char& |
| 3) char | 4) const char |
| 5) int | 6) const int |
| 7) all are equally good | 8) none are good |
- 9) What is the most appropriate way to fill the blank preceding the first formal parameter of Copy () in line 2?
- | | |
|-------------------------|------------------|
| 1) int& | 2) char& |
| 3) char | 4) const char |
| 5) int | 6) const int |
| 7) all are equally good | 8) none are good |
- 10) What is the most appropriate way to fill the blank preceding the second formal parameter of Copy () in line 2?
- | | |
|-------------------------|------------------|
| 1) int& | 2) char& |
| 3) char | 4) const char |
| 5) int | 6) const int |
| 7) all are equally good | 8) none are good |

- 11) You are writing a program to count the frequencies of characters that are read from a data file. (The computer uses the ASCII character set, which defines 255 different characters.) Which of the following array declarations is most appropriate, given that input characters will be used to index into the `freqCount` array?

- 1) `char freqCount[255];`
- 2) `char freqCount[int];`
- 3) `int freqCount[255];`
- 4) `int freqCount[char];`
- 5) none of these

- 12)** Which of the following statements about passing C++ arrays as function parameters is false?

- 1) It is impossible to pass an array by value.
- 2) When declaring an array as a formal parameter, you do not attach an ampersand (&) to the name of the component type.
- 3) When declaring an array as a formal parameter, you must include its size within square brackets.
- 4) At run time, the base address of the actual parameter is passed to the function.
- 5) By default, a single array element is passed by value.

- 13)** Which of the following cannot be used to store the string "Mary" into nameStr?

- 1) `char nameStr[5] = "Mary";`
- 2) `char nameStr[5];`
`nameStr = "Mary";`
- 3) `char nameStr[] = "Mary";`
- 4) `char nameStr[5];`
`strcpy(nameStr, "Mary");`
- 5) 2 and 3 cannot be used
- 6) all can be used
- 7) none can be used

- 14)** Given the declaration: `char myName[4] = "Ben";`

which of the following does not output "Ben" and only "Ben"?

- 1) `cout << myName;`
- 2) `for (i = 0; i < 3; i++)
 cout << myName[i];`
- 3) `for (i = 0; i < 3; i++)
 cout << myName;`
- 4) `i = 0;
while (myName[i] != '\0') {
 cout << myName[i];
 i++;
}`

- 5) none of these (they all output "Ben")

15) Given the declaration: `char message[10];`

which of the following statements is syntactically invalid?

- 1) `strcpy(message, "Welcome");`
- 2) `cin >> message;`
- 3) `message[2] = 'g';`
- 4) `if (strcmp(message, "Picnic") == 0)`
 `cout << "Hooray!";`
- 5) none of these (they are all valid)

16) Given the two lines of input data in an input file connected to an input file stream `iFile`:

```
Now is
the time.
```

what character is stored into the variable `someChar` by execution of the following code?

```
char str[20];
char someChar;

iFile >> str;
iFile.get(someChar);
```

- | | |
|--|------------------|
| 1) ' ' (a blank) | 2) 'i' |
| 3) '\n' | 4) 't' |
| 5) It cannot be determined from the information given. | 6) none of these |

17) Given the two lines of input data in an input file connected to an input file stream `iFile`:

```
Now is
the time.
```

what character is stored into the variable `someChar` by execution of the following code?

```
char str[20];
char someChar;

iFile.get(str, 5);
iFile.get(someChar);
```

- | | |
|--|------------------|
| 1) ' ' (a blank) | 2) 'i' |
| 3) '\n' | 4) 's' |
| 5) It cannot be determined from the information given. | 6) none of these |

Questions 18 through 20 refer to the following function:

```
bool Foo(const char Str1[], const char Str2[]) {  
    int Idx = 0;  
  
    while (Str1[Idx] != '\0' && Str2[Idx] != '\0') {  
        if (Str1[Idx] != Str2[Idx])  
            return false;  
        Idx++;  
    }  
    if (Str1[Idx] == '\0' && Str2[Idx] == '\0')  
        return true;  
    else  
        return false;  
}
```

18) What is the output of the following program fragment?

```
char Name1[20] = "Fred Flintstone";  
char Name2[20] = "Fred fLiNtstoNe";  
  
if ( Foo(Name1, Name2) )  
    cout << "Foo is true" << endl;  
else  
    cout << "Foo is false" << endl;
```

- | | |
|--|--------------------------------|
| 1) Foo is true | 2) Foo is false |
| 3) It cannot be answered from the information given. | 4) A runtime error would occur |
| 5) None of these | |

19) What is the output of the following program fragment? (Be careful here.)

```
char Name1[20] = "Fred Flintstone";  
char Name2[20] = "Fred G. Flintstone";  
  
if ( Foo(Name1, Name1) )  
    cout << "Foo is true" << endl;  
else  
    cout << "Foo is false" << endl;
```

- | | |
|--|--------------------------------|
| 1) Foo is true | 2) Foo is false |
| 3) It cannot be answered from the information given. | 4) A runtime error would occur |
| 5) None of these | |

20) What would happen if `Foo()` were called and one or both of its parameters did not include a null terminator?

- 1) Foo would definitely operate correctly
- 2) Foo would definitely process characters at logically invalid indices
- 3) Foo would possibly process characters at logically invalid indices
- 4) None of these