

Physics 234: Computational Physics

In-class Midterm Exam

Monday, February 14, 2011

Student's Name: _____

Fill-in-the-blank and multiple choice questions (20 points)

Mark your answers on the exam sheet in blue or black ink. Please be clear about your selections. Each question is worth one point unless otherwise indicated.

In code listings where the preamble is not written explicitly, you may assume the following.

```
#include <cassert>

#include <iostream>
using std::cout;
using std::cin;
using std::endl;
```

You may also assume that small snippets of code are properly embedded in a `main` function.

1. Rank these integer types in order of their size (from smallest to largest; write 1, 2, 3, 4 in the spaces provided).

___ short int
___ long int
___ unsigned char
___ const signed int

2. Which of the following is the correct scenario?

```
int a, b, c;  
int main()  
{  
    a = b = 5;  
    c = 3;  
    assert( a >= b > c );  
    return 0;  
}
```

- (a) program compiles, runs, and exits cleanly
(b) program compiles but fails at runtime on the `assert`
(c) program won't compile
3. Match each of the following literals with the most appropriate description. Write the corresponding letter in the space provided.
(a) `234u` (b) `2.99792E8F` (c) `2.99792x10^8` (d) `0x12` (e) `'\t'`

___ the number hexadecimal representation of 17
___ a char
___ an unsigned int
___ a single-precision floating-point number
___ just nonsense

4. Match each of the following object declarations with the most appropriate description. Write the corresponding letter in the space provided.

(a) `int* p;` (b) `float x;` (c) `char a;` (d) `char r[3];`

- ___ single-precision floating point number
___ integer type of exactly one byte in length
___ pointer to a fundamental type
___ 24 contiguous bits of storage

5. Write this program's output in the box provided.

```
const int N = 3;
for (int k = 2; k < N; ++k)
    for (int j = 1; j < k; ++j)
        for (int i = 0; i < j; ++i)
            cout << "(" << i << ", " << j << ", " << k << ")";
cout << endl;
```

terminal output:

6. What are the final values of `b` and `x` after the following two statements are executed. (Be sure to express your answers using a literal that is consistent with the type of each variable.)

```
int a = 3, b = -2;
const double x = a+-b---+4*3 / 5+a;
```

`b ==` `x ==`

7. What is the value of `(int)b`?

```
unsigned char a = 0;
const unsigned char b = --a;
```

- (a) 0
(b) -1
(c) 255
(d) 256

8. What value is assigned to `fx`?

```
const double x = 2.0;
const double fx = 1 + (1/3)*x + (8/3)*x*x;
```

- (a) 0.0
- (b) 1.0
- (c) 9.0
- (d) 12.333333333333333

9. (2 points) Suppose we use a C string

```
char config[] = "+++-----+++++-----++-----++++-----";
```

to encode the spin-up or spin-down character of a linear chain of local moments. We want to compose a function `flip` that ranges over the array and swaps each `char` from `'+'` to `'-'` and vice versa. Unfortunately, the programmer is confused and can't decide between these three attempts at writing the function body.

```
void flip(const char* const p)
{
    // while (*p) *p++ = (*p == '+') ? '-' : '+';
    // while (*p != '\0') { *p = '+' ? '-' : '+'; ++p; }
    /*
    for (int i = 0; p[i] ; ++i)
        if (p[i] = '+') p[i] = '-'; else p[i] = '+';
    */
    return;
}
```

Try to fix this function without adding anything new. Cross out as few `const` modifiers as you can get away with and uncomment whatever code you need to.

10. The following code compiles but generates a runtime error. Uncomment as few statements as possible so that the the program runs without crashing. (Just scratch out the offending `//` markers.)

```
double *p;
// double x = 1.0;
// int y = 2;
// p = &x;
// p += y;
*p = 2.0;
assert(*p == 2.0);
```

11. Consider a hypothetical integer type that is 4 bits wide (a *nybble*). Fill in all the missing entries in the table below. The second and third columns indicate the underlying bit pattern expressed in binary and hexadecimal, respectively. The first column (*) shows the corresponding *two's complement* interpretation as a decimal number. The last column (†) gives the *unsigned* interpretation.

<i>decimal*</i>	<i>binary</i>	<i>hex</i>	<i>decimal†</i>
	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
	0111	7	7
	1000	8	8
	1001	9	9
	1010		10
	1011		11
-4			12
			13
			14
	1111		15

12. Negate these three 8-bit two's complement numbers.

$$\boxed{0|1|0|1|1|0|1|0} \rightarrow \boxed{}$$

$$\boxed{0|0|0|0|0|0|0|0} \rightarrow \boxed{}$$

$$\boxed{1|0|0|0|0|0|0|0} \rightarrow \boxed{}$$

13. What is the best mathematical interpretation of the calculation carried out by the function call `myfunc(exp,sqrt,M_PI)`?

```
double myfunc( double (&f) (double),
               double (&g) (double), double y )
{
    return f(-3.0*g(((y+1.0)*y+2.0)*y+3.0));
}
```

- (a) $e^{-3\sqrt{1+\pi+2\pi^2+3\pi^3}}$
- (b) $\exp[-3(\pi^3 + \pi^2 + 2\pi + 3)^{1/2}]$
- (c) $e^{-3} \times e^{\sqrt{3+\pi(2+\pi(1+\pi))}}$
- (d) $\sqrt{\exp[-3(\pi^3 + \pi^2 + 2\pi + 3)]}$

14. What is the cheapest way to put the ninth power of `x` into a constant named `x9`?

- (a) `const double x9 = pow(x,9);`
- (b) `const double x9 = x*x*x*x*x*x*x*x*x;`
- (c) `const double x2 = x*x; const double x4 = x2*x2;`
`const double x8 = x4*x4; const double x9 = x8*x;`
- (d) `const double x2 = x*x; const double x3 = x2*x;`
`const double x9 = x3*x3;`

15. How many times will this loop execute on the lab computers?

```
for (unsigned long int k = 0; k != 1; --k);
```

- (a) zero times
- (b) once
- (c) over 4 billion times
- (d) indefinitely

16. What is the output from this program?

```
double inner_prod(const double *a0, const double *aN,
                  const double *b0)
{
    double sum = 0.0;
    while (a0 != aN) sum += *a0++**b0++;
    return sum;
}

int main()
{
    const double u[] = { 1, 2, 3, 4 };
    const double v[] = { 0, -1, 2, -3 };
    cout << inner_prod(u,u+4,v) << ", "
          << inner_prod(u,u+3,v+1) << endl;
    return 0;
}
```

- (a) 0,-6
- (b) 4,3
- (c) -8,-6
- (d) compiler error

17. This program uses the `inner_prod` function from question 16 to compute the matrix product of A and B. The variables `i` and `j` run over the rows and columns of C, where the result is stored. Choose the best description of how the data is packed into arrays.

```
const double A[N*(N+1)/2] = { /* some initial values */ };
const double B[N*(N+1)/2] = { /* some initial values */ };
double C[N*N];
for (int i = 0; i < N; ++i)
    for (int j = 0; j < N; ++j)
    {
        const double* const a = A + i*(i+1)/2;
        const double* const b = B + j*(j+1)/2;
        C[i+j*N] = inner_prod(a,a+1+( i < j ? i : j),b);
    }
```

- (a) A is a lower-triangular matrix packed along its rows
 - (b) B is an upper-triangular matrix packed along its rows
 - (c) B is a low-triangular matrix packed along its columns
 - (d) C is stored in row-major order.
18. (2 points) Given the C array definitions,

```
char p[] = { 'm', 'a', 's', 's', ' ', 'm', '=', '1', '.', '0', '\0' };
char q[] = "mass m = 1.0";
```

what do each of the following equalities evaluate to? In each case, circle either true (t), false (f), or unpredictable (u).

- (t / f / u) `p == q;`
- (t / f / u) `*p != *q;`
- (t / f / u) `*(p+10) < *(q+10);`
- (t / f / u) `*(p+6) == *(q+6);`