Name: Plaide SHEMA

CS Login: Pskyma01

COMP11

Midterm – Fall 2023

This test is closed book. We suggest that you read the entire test before starting, and tackle the questions that you know how to do first. Some harder questions are worth just a few points – if you get stuck, do them last.

Question 1: Binary (8 points)

Convert the following binary numbers into decimal:

1.) `1011` $= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= 8 + 0 + 2 + 1$

$= 11$

2.) `10110` $= (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

$= 16 + 0 + 4 + 2 + 0$

$= 22$

Convert the following decimal numbers into binary:

1.) `24` $=$ ~~24 2~~

$$\begin{array}{c|c} 24 & 0 \\ 12 & 0 \\ 6 & 0 \\ 3 & 1 \\ 1 & 1 \end{array} \uparrow$$

$24 = 11000$

2.) `15` $= 01111$

Question 2: Loops and Arrays (24 points)

In each problem, an array (top) is modified by a loop (middle). Fill in the resulting array values (bottom). You can assume that the array is called `arr` and its length is stored as an int, `length`.

1.)

A	B	C	D	E	F	G	H	I	J

```
for(int i = 1; i < length; i++){
      arr[i - 1] = arr[i];
}
```

B	C	D	E	F	G	H	I	J	J

2.)

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9

```
for(int i = length-1; i > 0; i--){
     if(arr[i] <= arr[length - 1]){
           arr[i - 1] = arr[i];
     }
}
```

B	C	D	E	F	G	H	I	J	J

3.)

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9

```
for(int i = 0; i < length; i++){
      if(i % 2 == 0){
           arr[i] = 'z';
      }
}
```

Z	B	Z	D	Z	F	Z	H	Z	J

↓
Z

4.)

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9

```
for(int i = length; i > 1; i-=2){
      arr[i - 2] = arr[i - 1];
}
```

B	B	D	D	F	F	H	H	J	J

5.)

A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9

```
for(int i = 0; i < length; i++){
      if(arr[i] > arr[2] + 2){
           arr[i] = 'x';
      }
}
```

A	B	C	D	E	X	X	X	X	X

6.)

A	B	C	D	E	F	G	H	I	J

```
for(int i = 0; i < length; i++){
      int idx = length - 1 - i;
      arr[i] = arr[idx];
      if(i > length/2){
           break;
      }
}
```

J	I	H	G	F	E	G	H	I	J

Question 3: Memory (28 points)

Consider the following program (assume the necessary `#include` and `using` statements are above):

```
int main(int argc, char *argv[]){
   int length;
   int running_total = 0;
   int *numbers = readNums(argv[1], &length);

   for(int j = 0; j < length; j++){
        running_total += numbers[j];
        cout << running_total << endl;
   }

   return 0;
}

int *readNums(string filename, int *num_count){
   int count;
   ifstream infile(filename);

   infile >> count;
   int *nums = new int[count];
   for(int i = 0; i < count; i++){
        infile >> nums[i];
   }

   infile.close();
   *num_count = count;
   return nums;
}
```

1.) List every variable that holds the address of another variable.

- Valiable numbers and nums.

2.) Circle the variable(s) that point to an address on the heap

num_count   length   count   [✓ nums]   running_total

3.) Circle the variable(s) contained within the first frame that will be pushed onto the stack during the execution of this program.

numbers   count   infile   [✓ running_total]   [✓ j]   [✓ i]

4.) How many stack frames will be pushed onto the stack during the execution of this program?

- Four.

5.) What code would need to be added to this program for valgrind to run without errors and definitely lost memory? Where should it be added (which line number)?

delete []numbers , before return 0; / line 16

6.) Imagine that this program is passed a file that contains the following information:

```
3
3
9
4
```

What would the output of the program be?

3
6
15
19

7.) In general, when does memory need to be allocated on the heap?

- The memory need to be allocated during the declaration.

Question 4: Booleans (20 points) ✓

What color(s) will each of the following code snippets print?

```
bool b = true;
bool a = false;

if(b && !a){
    cout << "pink" << endl;
}else{
    cout << "yellow" << endl;
}
```

Output: pink

```
int z = 5, y = 18;
bool c = false;
bool d = !c;
bool e = (z == y) || (z < y);

if(c || e || d){
    cout << "green" << endl;
}else{
    cout << "red" << endl;
}
```

Output: red.

```
int a = 4, b = 16;

if( (a * a) == b ){
    if( !(b < 20) ){
        cout << "blue" << endl;
    }else{
        cout << "red" << endl;
    }
}else{
    if( !(b < 20) )
    {
        cout << "pink" << endl;
    }else{
        cout << "green" << endl;
    }
}
```

Output: red

```
int a = 4, b = 16;
bool c = !((b / a) == (2 * 2));
bool d = (c || (( a + 4 ) < b));

if ( !d ){
    cout << "pink" << endl;
}else{
    if( !(b < 20) ){
        cout << "yellow" << endl;
    }else{
        cout << "blue" << endl;
    }
}
```

Output: blue.

```
int x = 12 - 5;
int y = x * 3 - 2;
int z = (y - x - 1) / 6;

if( z == 2 ){
    if( ( z + 3 ) < 10 ){
        cout << "red" << endl;
    }else{
        cout << "yellow" << endl;
    }
}else{
    if( ( z + 3 ) < 10 ){
        cout << "blue" << endl;
    }else{
        cout << "pink" << endl;
    }
    cout << "green" << endl;
}
```

Output: ~~blue~~. Red.

Question 5: Structs and Pointers (20 points) ✓

Consider the following struct definitions:

```
struct A;
struct B;

struct A {
    int value;
    B *linkB;
};

struct B {
    int value;
    A *linkA;
};
```

Next, follow the lines of code below and fill in each square with either a (large) dot or an arrow to indicate how the field of each struct must be accessed in order for the program to compile.

```
A s1;
A *s2 = &s1;
B s3;
```

```
s1____●____value = 12;

s2____->____linkB = &s3;

s1____->____linkB____->____linkA____●____value = 32;

(*(s1____●____linkB))____->____linkA = &s1;

s3____->____linkA = s2____->____linkB____->____linkA;
```