COP 3514-001 – Program Design
Test 2

86+6=92

Name: Steven Romeiro

100

U-Number:

U16803837

int n = size of array

Question 1 (20 points). Write a function that takes in an argument "`const int *a`" (an array of integers) of length n. This function will search the array and find the largest and second largest elements and store them into pointed variables. Use pointer arithmetic and no subscripting.

```
Void search (const int * a, int n, int *largest, int *SecondLargest)
  {
    const int *p;
    *largest = *SecondLargest = *a;
    for(p=a+1; p< a+n; p++)
    {
      if( *largest < *p)
        {
          *largest = *p;
        }
      else if( *SecondLargest<*p || *SecondLargest >=*largest)
        {
          *SecondLargest = *p;
        }
  }
```

20

+20

Question 2 (20 points). Write a program (`main` and everything) that adds up command line arguments which are assumed to be integers. The inputs "`sum 8 1 62`" would print "`Total: 94`"

```
int main (int argc, char *argv[])
{
    int i = 1, sum = 0;
    for (i; i < argc; i++)
    {
        sum += atoi(argv[i]);
    }
    printf ("Total: %d", sum);
    return 0;
}
```

20

+20

Question 3 (6 points). What is wrong with this code snippet?

```
float *test_function(void) {
      float f = 1.0f;
      f = f + 1;
      return &f;
}
```

Once this function terminates, the automatic storage for float f will be given up. function will return memory address to given up space. Not good.

+6

Question 4 (6 points). What is the similarities/differences between a Union and Structure?

Unions & Structures both allow for storing data of different types.

+6

They differ in that unions allocate memory for the largest data type present & the members share that memory space, while structures allocate memory for each member separately. Unions should be used to store only one type of data at a time since memory will be overwritten. Structures can store all the data for each member at once

+12

Question 5 (12 points, 2 points each). Which of these expressions are equivalent if a is an array of ints size 10, p is a pointer, p = a? Write 1 for equivalent and 0 for not.

(a) `p == a[0]`

0

(b) `p == &a[0]`

1

(c) `*p == a[0]`

1

(d) `p[0] == a[0]`

1

(e) `*p + 1 == *&a[1]`

0

(f) `p + 3 == &a[3]`

1

12

Question 6 (12 points, 2 points each). Which of the following statements are valid C code if number is an int, p is a pointer, and q is a pointer? Which will result in the possibility of illegal memory access? If not valid, give a reason why. Remember, something can be valid and still illegally memory access.

(a) `p = number;`

not valid. incorrect types. Pointer P expects address but got int number instead

(b) `*p = &number;`

not valid. Value at pointer P should be integer but got address. illegal memory access

(c) `&p = q;`

not valid. Correct assignment should be p=q. P is L-value illegal memory access.

(d) `p = &q;`

Not valid unless p is pointer to a pointer. illegal memory access.

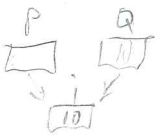
(e) `p = *&q;`

Valid.

(f) `p = *q;`

not valid. pointer p expects address but received integer value instead.

4



16

Question 7 (12 points, 3 each). Defining what is happening in each expression below, is the increment happening before or after the value is retrieved?

`*p++`

increment p then dereference. Associativity right to left

`(*p)++`

dereference p then increment. Parenthesis has highest order

`*++p`

increment p then dereference. Associativity right to left

`++*p`

dereference p then increment. Associativity right to left

12

+12

Question 9 (6 points). Give a short description of strlen, strcmp, strcat, strcpy.

- strlen is a function that returns how many characters are in the string argument passed to it not counting the null character.
- Strcmp compares two strings for equality based on length & ASCII code value. returns -1, 0, 1 when first string is less than, equal to, greater than second string respectively.
- Strcat concatenates string one with string two that is provided by user.
- Strcpy accepts two string arguments & copies the second string into the variable holding the first string.

BONUS QUESTION (3 Points). What is Stack-based memory allocation? How does it relate to local variables in C?

Stack based mem allocation is how C handles local variables. It requests a stack of memory & assigns these variables to that stack. Keeping them "close" or in the same area for retrieval

3

BONUS QUESTION (3 Points). What is memory segmentation?

memory segmentation is when the computer segments or splits up memory & reserves it for a needed process.

3