Use the program below for problems 1 – 3

```c
#include<stdio.h>

#define SIZE 5

void initArray(int[]);

int main()
{
    int x[SIZE] = {1, 3, 5};
    printf("%d ", x[2]);
    initArray(x);
    printf("%d
", x[2]);
    return(0);
}

void initArray(int y[])
{
    int lcv;
    for(lcv = 0; lcv < SIZE; lcv++)
    {
        y[lcv] = lcv + 1;
    }
}
```

1. What is the output of the the program provided on the left?
   A. 5 5
   B. 3 3
   C. 5 3
   D. None of the above

2. Which of the following is stored in the array at index 3 before the call to the `initArray` function?
   A. 0
   B. 5
   C. 7
   D. None of the above

3. Which of the following would NOT be a possible outcome of changing the relational operator in the loop control expression on line 24 from less than to less than or equal to?
   A. The compiler would issue a warning.
   B. The program will crash.
   C. The output will be unexpected.
   D. None of the above

4. Which of the following statements regarding arrays is FALSE?
   A. All of the elements in an array in C share the same data type.
   B. All of the elements in an array in C share a single variable name, but a different index value.
   C. An array element, in C, cannot be accessed if it has not been initialized.
   D. None of the above

Use the code below for problems 5 – 6

```c
#define EXIT -1
#define MAXSIZE 10

int getData(int x[])
{
    int lcv = 0;
    do
    {
        printf("Enter data #\%d: ", lcv + 1);
        scanf("%d", &x[lcv]);
        lcv++;
    }while(lcv < MAXSIZE && x[lcv - 1] != EXIT);
    return(x[lcv - 1] != EXIT ? lcv : lcv – 1);
}
```

5. Which of the following is returned from the function on the left if 10 non-negative integer values have been input by the user?
   A. 6
   B. 7
   C. 10
   D. None of the above

6. Which of the following is returned from the function above if 9 non-negative integer values have been entered before the user enters the EXIT value?
   A. 8
   B. 9
   C. 10
   D. None of the above
Use the code segment below for problems 7 – 8

```c
int temp;
int lcv;
int array[SIZE] = {8, 4, 1, 2, 7, 5};

for(lcv = 0; lcv < SIZE / 2; lcv++)
{
    temp = a[SIZE - 1 - lcv];
    a[SIZE - 1 - lcv] = a[lcv];
    a[lcv] = temp;
}
```

7. Given that SIZE has been defined as 6, which of the following is found in index 3 after the code above has been executed?
   A. 2  
   B. 1  
   C. 7  
   D. 4

8. Which of the following would be found in index of the array 3 if the loop control expression was changed from `lcv < SIZE / 2` to `lcv < SIZE` after the code segment above was executed?
   A. 2  
   B. 1  
   C. 7  
   D. 4

Use the code below for problems 9 – 10

```c
int x[5] = {3, 4, 5, 6, 7};
int lcv;
int temp;

temp = x[0];

for(lcv = 0; lcv < 4; lcv++)
{
    x[lcv] = x[(lcv + 1) % 5];
}

x[lcv] = temp;

printf("x[0] = %d\n", x[0]);
printf("x[4] = %d\n", x[4]);
```

9. What is the output of the first print statement above?
   A. x[0] = 3  
   B. x[0] = 4  
   C. x[0] = 5  
   D. None of the above

10. What is the output of the second print statement above?
    A. x[4] = 3  
    B. x[4] = 6  
    C. x[4] = 7  
    D. None of the above

11. Given an initial array of {6, 4, 5, 2, 3, 1} how many passes are required before the array is sorted using the bubble sorting algorithm? Note that this question refers to the configuration of the array specified and not any general six-element array. The sorted array would be {1, 2, 3, 4, 5, 6}.
    A. A minimum of four passes are required for the array to be sorted.
    B. A minimum of five passes are required for the array to be sorted.
    C. A minimum of six passes are required for the array to be sorted.
    D. None of the above
12. Given an initial array of \{3, 6, 4, 1, 5, 2\} which of the following represents the order of the array after two passes through the selection sort algorithm?
   A. \{3, 2, 4, 1, 5, 6\}
   B. \{3, 4, 1, 2, 5, 6\}
   C. \{3, 2, 1, 4, 5, 6\}
   D. None of the above

13. Given the array below, and a target value of 17, which of the following are the values of the \textit{mid} variable as the binary search progresses?
   \[\text{int } x[9] = \{3, 7, 9, 11, 13, 14, 15, 17, 21\};\]
   A. 4, 7
   B. 4, 6, 7
   C. 5, 7
   D. None of the above

14. Given the array below, which of the following is the result of the print statement that follows?
   \[\text{int } x[9] = \{3, 7, 9, 11, 13, 14, 15, 17, 21\};\]
   \[\text{printf("\%d", x);}\]
   A. The print statement will display only the first element of the array.
   B. The print statement will display only the final element of the array.
   C. The print statement will display all of the values inside of the array.
   D. None of the above

\textbf{Use the code below for problems 15 – 16}

\[\text{int } x[4] = \{25, 9, 15, 17\};\]
\[\text{int } j;\]
\[\text{int } k;\]
\[\text{int } sum = 0;\]
\[\text{for}(j = 0; j < 4; j++)\]
\[\{\]
\[\text{for}(k = 3; k < x[j] / 2; k += 2)\]
\[\{\]
\[\text{if}(x[j] \% k == 0)\]
\[\{\]
\[\text{sum } += k;\]
\[k = x[j];\]
\[\}\]
\[\}\]
\[\text{printf("sum } = \%d\n", sum);}\]
\[\text{printf("j } = \%d k } = \%d\n", j, k);}\]

15. What is the output of the first print statement in the code above?
   A. sum = 17
   B. sum = 13
   C. sum = 11
   D. None of the above

16. What is the output of the second print statement in the code above?
   A. j = 4 k = 7
   B. j = 3 k = 7
   C. j = 4 k = 9
   D. None of the above
17. Given that the variable `scores` is an array, which of the following function calls utilizes pass by address?
   A. `printData(scores);`
   B. `printData(scores[3], scores[4], scores[5]);`
   C. `printData(scores[4]);`
   D. Both A and B

```c
#include<stdio.h>
#include<string.h>

int main()
{
    char str[15] = "Good Bye";
    char str2[15] = "hello";

    printf("Result #1: %d\n", strcmp(str, str2));
    str[strlen(str2)] = str2[0];
    printf("Result #2: %s\n", str);
    return(0);
}
```

18. Which of the following describes the integer value output of the first print statement?
   A. A positive value.
   B. A negative value.
   C. A zero value.
   D. None of the above

19. What is the output of the second print statement?
   A. Result #2: GoodhBye
   B. Result #2: Good hye
   C. Result #2: Good
   D. None of the above

20. Which of the following is NOT a part of the CPU instruction cycle?
   A. Fetch Instructions
   B. Process Data
   C. Write Data to Memory
   D. None of the above

Use the code below for problems 21 – 23

```c
int x = 3;
int y = 4;
int z = 6;

z -= x + y;
printf("z = %d\n", z);  // 21.

x = 718 % y;
printf("x = %d\n", x);  // 22.

printf("result = %.2f\n", 3.0 / y);
```

21. What is the output of the first print statement of the code on the left?
   A. z = -1
   B. z = 7
   C. z = 5
   D. None of the above

22. What is the output of the second print statement of the code on the left?
   A. x = 0
   B. x = 1
   C. x = 2
   D. x = 3

23. What is the output of the third print statement of the code above?
   A. result = 0.00
   B. result = 0.75
   C. result = 1.00
   D. None of the above
Use the program below for problems 24 – 25

```c
#include<stdio.h>

void initValues(int*, int*, int*);

int remainder(int);

int main()
{
    int x = 0;
    int y = 0;
    int z = 0;
    int rem;

    initValues(&x, &y, &z);

    printf("x = %d y = %d z = %d\n", x, y, z);

    rem = remainder(x + y + z);

    printf("Remainder = %d\n", rem);

    return(0);
}

void initValues(int *y, int *z, int *x)
{
    *y = 4;
    *z = 2;
    *x = 9;
}

int remainder(int sum)
{
    return(sum++ % 3);
}
```

24. What is the output of the first print statement in the program above?
   A. x = 9 y = 4 z = 2 
   B. x = 2 y = 9 z = 4 
   C. x = 4 y = 2 z = 9 
   D. None of the above

25. What is the output if the second print statement in the program above?
   A. Remainder = 0 
   B. Remainder = 1 
   C. Remainder = 2 
   D. None of the above

Use the code below for problem 26

```c
int sum = 0;
int c = 543;
int d = 2;

while(c > 0)
{
    sum += (int)pow(c % 10, d);
    printf("sum = %d\n", sum);
    c /= 10;
}
```

26. Which of the following is NOT a line of output produced by the code segment above?
   A. sum = 9 
   B. sum = 16 
   C. sum = 25 
   D. None of the above
Use the code below for problems 27 – 30

```c
int a = 3;
int b = 0;
int c = 0;

printf("result #1: %d\n", b == a || c);
printf("result #2: %d\n", a > 3 || c != -3 && b);
printf("result #3: %d\n", ++b || ++c || a++);
printf("a b c: %d %d %d\n", a, b, c);
```

27. What is the output of the first print statement above?
   A. result #1: 1
   B. result #1: 0
   C. result #1: 3
   D. None of the above

28. What is the output of the second print statement above?
   A. result #2: 1
   B. result #2: 0
   C. result #2: 3
   D. None of the above

29. What is the output of the third print statement above?
   A. result #3: 1
   B. result #3: 0
   C. result #3: 3
   D. None of the above

30. What is the output of the fourth print statement above?
   A. a b c: 4 1 1
   B. a b c: 3 1 1
   C. a b c: 3 1 0
   D. None of the above

Use the code segment below for problems 31 – 32

```c
int lcv;
int ct1 = 0;
int ct2 = 0;

for(lcv = 0; lcv < 20; lcv++)
{
    switch(lcv % 5)
    {
    case 4:
    case 1:
        ct1++;
        break;
    default:
        ct2++;
    }
}

printf("ct1 = %d\n", ct1);
printf("ct2 = %d\n", ct2);
```

31. What is the output of the first print statement above?
   A. ct1 = 8
   B. ct1 = 7
   C. ct1 = 6
   D. None of the above

32. What is the output of the second print statement above?
   A. ct2 = 12
   B. ct2 = 13
   C. ct2 = 14
   D. None of the above
Use the code below for problems 33 – 34

```c
int row;
int col;
int sz = 3;

for(row = 1; row <= sz; row++)
{
    for(col = 1; col <= sz; col++)
    {
        col >= sz - row + 1 ? printf("%d", sz) : printf("%c", '¥');
    }
    printf("\n");
}
```

33. How many times is the value of the variable `sz` displayed as a result of executing the code segment above?
A. 3  
B. 6  
C. 9  
D. None of the above

34. What is the total number of visible characters displayed as a result of executing the code segment above?
A. 6  
B. 9  
C. 12  
D. None of the above

Use the function below for problems 35 – 36

```c
int calcFactor(int wages, int max)
{
    int factor;

    factor = wages / max;

    return((factor + 2) % (factor + 1));
}
```

35. Given that wages = 7 and max = 5, what is the value returned from the function above?
A. 0  
B. 1  
C. 2  
D. None of the above

36. Given that wages = 4 and max = 3, what is the value returned from the function above?
A. 0  
B. 1  
C. 2  
D. None of the above

Individual Assignments

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework:</td>
<td>30.4</td>
</tr>
<tr>
<td>Lab Quiz:</td>
<td>48.8</td>
</tr>
<tr>
<td>Lecture Quiz:</td>
<td>8.0</td>
</tr>
<tr>
<td>Exam Total:</td>
<td>252</td>
</tr>
</tbody>
</table>

37. Which of the following would produce the same format as the lab quiz line of output as seen on the left?
A. `printf("Lab Quiz: %14.1f\n", labQuiz);`
B. `printf("Lab Quiz: %13.1f\n", labQuiz);`
C. `printf("Lab Quiz: %12.1f\n", labQuiz);`
D. None of the above

38. Which of the following would produce the same format as the exam total line of output as seen above?
A. `printf("Exam Total: %lld\n", final);`
B. `printf("Exam Total: %10d\n", final);`
C. `printf("Exam Total: %12d\n", final);`
D. None of the above
Use the code segment below for problems 39 – 41

```c
int n = 13;
while(n > 1)
{
    if(n % 2 == 0)
    {
        n = n / 2;
    }
    else
    {
        n = n * 3 + 1;
    }
    printf(" %d ", n);
}
```

39. What is the first value of the output produced in the code segment above?
   A. 13  
   B. 39  
   C. 6   
   D. None of the above

40. What is the last value of the output produced in the code segment above?
   A. 1   
   B. 2   
   C. 0   
   D. None of the above

41. How many total values are produced as output by the code segment above?
   A. 9   
   B. 8   
   C. 7   
   D. None of the above

42. Given that YEAR has been defined as 2010 (#define YEAR 2010) which of the following would
    produce the expected output below?

    Semester: Spring 2010

   A. printf("Semester: Spring YEAR\n");
   B. printf("Semester: Spring %d\n", YEAR);
   C. Both A and B
   D. None of the above

float area;
int r = 2;
area = 4 / 3 * M_PI * r * r;

43. Given the statements above, which of the following print statements would produce a value that is
    different from the other two?
   A. printf("Area = %.2f\n", 0.0 + area);
   B. printf("Area = %.2f\n", (float)area);
   C. printf("Area = %.2f\n", area);
   D. All three of the above print statements produce the same value.

44. Given the statements above, which of the following would assign the same value to the variable area as
    the original statement above?
   A. area = (float) 4 / 3 * M_PI * r * r;
   B. area = M_PI * 4 / 3 * r * r;
   C. area = (float) 4 / 3 * M_PI * pow(r, 2);
   D. None of the above
Use the program below for problem 45

```c
#include<stdio.h>
#include<string.h>
#include<ctype.h>

#define SIZE 51

void rot13(char[], int);

int main()
{
    char str[SIZE] = "Hello";
    rot13(str, strlen(str) - 1);
    printf("\n");
    return(0);
}

void rot13(char x[], int i)
{
    if(i >= 0)
    {
        if(isalpha(x[i]))
        {
            x[i] += toupper(x[i]) <= 'M' ? 13 : -13;
        }
        printf("%c", x[i]);
        rot13(x, i - 1);
    }
}
```

45. What is the output generated by the program above?
   A. Uryyb  
   B. URYYB  
   C. byyrU  
   D. BYYRU

Use the code segment below for problems 46 – 47

```c
int *a;
int *b;
int c = 5;
int d = 2;

a = &c;
b = a;

printf("Result: %d\n", *a + *b);

a = &d;

printf("Result #2: %d\n", *b - *a);
```

46. What is the output of the first print statement?
   A. Result: 5  
   B. Result: 7  
   C. Result: 10  
   D. None of the above

47. What is the output of the second print statement?
   A. Result #2: 0  
   B. Result #2: 3  
   C. Result #2: -3  
   D. None of the above
48. Which of the following characters are used within a MATLAB fscanf placeholder to indicate that a data item of a particular data type is present in a file but is not to be read but rather skipped?

A. ^
B. ~
C. *
D. None of the above

49. Which of the following correctly closes a file handle variable \( fh \) that is currently connected to an external data file called \( labData \)?

A. \( fh = fclose(labData); \)
B. \( fclose(labData); \)
C. \( fclose(fh); \)
D. None of the above

50. Which of the following is TRUE?
A. Grades in CS 159 are based only on the number of points earned in the class. Matters not related to the class such as scholarship or academic probation status cannot be taken into consideration when determining grades.
B. With a class as large as CS 159 there will always be 10-12 students who just miss a cutoff. To move the cutoff just a few more points only introduces another 10-12 students who just miss the revised cutoff and request the same consideration that was given to the first 10-12 students.
C. Requests for re-grades on assignments posted or returned more than 5 days ago will be denied. Check all relevant end of semester announcements on the front of this exam and as posted on Blackboard Monday or Tuesday of next week.
D. All of the above.
This page lists C operators in order of *precedence* (highest to lowest). Their *associativity* indicates in what order operators of equal precedence in an expression are applied.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>Parentheses (function call) (see Note 1)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>[]</td>
<td>Brackets (array subscript)</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>Member selection via object name</td>
<td></td>
</tr>
<tr>
<td>-&gt;</td>
<td>Member selection via pointer</td>
<td></td>
</tr>
<tr>
<td>++ --</td>
<td>Postfix increment/decrement (see Note 2)</td>
<td></td>
</tr>
<tr>
<td>++ --</td>
<td>Prefix increment/decrement</td>
<td>right-to-left</td>
</tr>
<tr>
<td>+ -</td>
<td>Unary plus/minus</td>
<td>left-to-right</td>
</tr>
<tr>
<td>! ~</td>
<td>Logical negation/bitwise complement</td>
<td>left-to-right</td>
</tr>
<tr>
<td>(type)</td>
<td>Cast (change type)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>* &amp; sizeof</td>
<td>Dereference/Bitsize determination</td>
<td>left-to-right</td>
</tr>
<tr>
<td>/ %</td>
<td>Multiplication/division/modulus</td>
<td>left-to-right</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition/subtraction</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&lt;&lt; &gt;&gt;</td>
<td>Bitwise shift left/right</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&lt; &lt;= &gt; &gt;=</td>
<td>Relational less than/greater than or equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>== !=</td>
<td>Relational is equal to/is not equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&amp;</td>
<td>Bitwise AND</td>
<td>left-to-right</td>
</tr>
<tr>
<td>^</td>
<td>Bitwise exclusive OR</td>
<td>left-to-right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bitwise inclusive OR</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Logical AND</td>
<td>left-to-right</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>? :</td>
<td>Ternary conditional</td>
<td>right-to-left</td>
</tr>
<tr>
<td>=</td>
<td>Assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>+= -=</td>
<td>Addition/subtraction assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>*= /=</td>
<td>Multiplication/division assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>%= &amp;=</td>
<td>Modul mode/bitwise AND assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>^=</td>
<td>=</td>
<td>Bitwise exclusive/inclusive OR assignment</td>
</tr>
<tr>
<td>&lt;&lt;= &gt;&gt;=</td>
<td>Bitwise shift left/right assignment</td>
<td>right-to-left</td>
</tr>
<tr>
<td>,</td>
<td>Comma (separate expressions)</td>
<td>left-to-right</td>
</tr>
</tbody>
</table>
## ASCII Table

<table>
<thead>
<tr>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
<th>Char</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>(null)</td>
<td>0</td>
<td>space</td>
<td>32</td>
<td>@</td>
<td>64</td>
<td>`</td>
<td>96</td>
</tr>
<tr>
<td>(soh)</td>
<td>1</td>
<td>!</td>
<td>33</td>
<td>A</td>
<td>65</td>
<td>a</td>
<td>97</td>
</tr>
<tr>
<td>(stx)</td>
<td>2</td>
<td>&quot;</td>
<td>34</td>
<td>B</td>
<td>66</td>
<td>b</td>
<td>98</td>
</tr>
<tr>
<td>(etx)</td>
<td>3</td>
<td>#</td>
<td>35</td>
<td>C</td>
<td>67</td>
<td>c</td>
<td>99</td>
</tr>
<tr>
<td>(eot)</td>
<td>4</td>
<td>$</td>
<td>36</td>
<td>D</td>
<td>68</td>
<td>d</td>
<td>100</td>
</tr>
<tr>
<td>(enq)</td>
<td>5</td>
<td>%</td>
<td>37</td>
<td>E</td>
<td>69</td>
<td>e</td>
<td>101</td>
</tr>
<tr>
<td>(ack)</td>
<td>6</td>
<td>&amp;</td>
<td>38</td>
<td>F</td>
<td>70</td>
<td>f</td>
<td>102</td>
</tr>
<tr>
<td>(bel)</td>
<td>7</td>
<td></td>
<td>39</td>
<td>G</td>
<td>71</td>
<td>g</td>
<td>103</td>
</tr>
<tr>
<td>(bs)</td>
<td>8</td>
<td>(</td>
<td>40</td>
<td>H</td>
<td>72</td>
<td>h</td>
<td>104</td>
</tr>
<tr>
<td>(ht)</td>
<td>9</td>
<td>)</td>
<td>41</td>
<td>I</td>
<td>73</td>
<td>i</td>
<td>105</td>
</tr>
<tr>
<td>(nl)</td>
<td>10</td>
<td>*</td>
<td>42</td>
<td>J</td>
<td>74</td>
<td>j</td>
<td>106</td>
</tr>
<tr>
<td>(vt)</td>
<td>11</td>
<td>+</td>
<td>43</td>
<td>K</td>
<td>75</td>
<td>k</td>
<td>107</td>
</tr>
<tr>
<td>(np)</td>
<td>12</td>
<td>,</td>
<td>44</td>
<td>L</td>
<td>76</td>
<td>l</td>
<td>108</td>
</tr>
<tr>
<td>(cr)</td>
<td>13</td>
<td>-</td>
<td>45</td>
<td>M</td>
<td>77</td>
<td>m</td>
<td>109</td>
</tr>
<tr>
<td>(so)</td>
<td>14</td>
<td>.</td>
<td>46</td>
<td>N</td>
<td>78</td>
<td>n</td>
<td>110</td>
</tr>
<tr>
<td>(si)</td>
<td>15</td>
<td>/</td>
<td>47</td>
<td>O</td>
<td>79</td>
<td>o</td>
<td>111</td>
</tr>
<tr>
<td>(dle)</td>
<td>16</td>
<td>0</td>
<td>48</td>
<td>P</td>
<td>80</td>
<td>p</td>
<td>112</td>
</tr>
<tr>
<td>(dc1)</td>
<td>17</td>
<td>1</td>
<td>49</td>
<td>Q</td>
<td>81</td>
<td>q</td>
<td>113</td>
</tr>
<tr>
<td>(dc2)</td>
<td>18</td>
<td>2</td>
<td>50</td>
<td>R</td>
<td>82</td>
<td>r</td>
<td>114</td>
</tr>
<tr>
<td>(dc3)</td>
<td>19</td>
<td>3</td>
<td>51</td>
<td>S</td>
<td>83</td>
<td>s</td>
<td>115</td>
</tr>
<tr>
<td>(dc4)</td>
<td>20</td>
<td>4</td>
<td>52</td>
<td>T</td>
<td>84</td>
<td>t</td>
<td>116</td>
</tr>
<tr>
<td>(nak)</td>
<td>21</td>
<td>5</td>
<td>53</td>
<td>U</td>
<td>85</td>
<td>u</td>
<td>117</td>
</tr>
<tr>
<td>(syn)</td>
<td>22</td>
<td>6</td>
<td>54</td>
<td>V</td>
<td>86</td>
<td>v</td>
<td>118</td>
</tr>
<tr>
<td>(etb)</td>
<td>23</td>
<td>7</td>
<td>55</td>
<td>W</td>
<td>87</td>
<td>w</td>
<td>119</td>
</tr>
<tr>
<td>(can)</td>
<td>24</td>
<td>8</td>
<td>56</td>
<td>X</td>
<td>88</td>
<td>x</td>
<td>120</td>
</tr>
<tr>
<td>(em)</td>
<td>25</td>
<td>9</td>
<td>57</td>
<td>Y</td>
<td>89</td>
<td>y</td>
<td>121</td>
</tr>
<tr>
<td>(sub)</td>
<td>26</td>
<td>;</td>
<td>58</td>
<td>Z</td>
<td>90</td>
<td>z</td>
<td>122</td>
</tr>
<tr>
<td>(esc)</td>
<td>27</td>
<td>:</td>
<td>59</td>
<td>{</td>
<td>91</td>
<td>}</td>
<td>123</td>
</tr>
<tr>
<td>(fs)</td>
<td>28</td>
<td>&lt;</td>
<td>60</td>
<td>\</td>
<td>92</td>
<td></td>
<td>124</td>
</tr>
<tr>
<td>(gs)</td>
<td>29</td>
<td>=</td>
<td>61</td>
<td>]</td>
<td>93</td>
<td>}</td>
<td>125</td>
</tr>
<tr>
<td>(rs)</td>
<td>30</td>
<td>&gt;</td>
<td>62</td>
<td>^</td>
<td>94</td>
<td>~</td>
<td>126</td>
</tr>
<tr>
<td>(us)</td>
<td>31</td>
<td>?</td>
<td>63</td>
<td>_</td>
<td>95</td>
<td>(del)</td>
<td>127</td>
</tr>
<tr>
<td>Lab Time</td>
<td>Lab Day</td>
<td>Location and TA</td>
<td>Section</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>--------------------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Tuesday</td>
<td>SC 189 – Bryan Ehrlich</td>
<td>0101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Tuesday</td>
<td>SC 289 – Jacob Harvey</td>
<td>0102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30</td>
<td>Tuesday</td>
<td>SC 189 – Mike James</td>
<td>0201</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Tuesday</td>
<td>SC 189 – Caleb Bean</td>
<td>0301</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Tuesday</td>
<td>SC 189 – Kyle Egbert</td>
<td>0401</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Tuesday</td>
<td>SC 289 – Ashley Frake</td>
<td>0402</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Tuesday</td>
<td>SC 189 – Jessica Sparks</td>
<td>0501</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Wednesday</td>
<td>SC 189 – Xi Luo</td>
<td>0601</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Wednesday</td>
<td>SC 289 – Joel Wesseling</td>
<td>0602</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30</td>
<td>Wednesday</td>
<td>SC 189 – Ryne Rayburn</td>
<td>0701</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Wednesday</td>
<td>SC 189 – Mike James</td>
<td>0801</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Wednesday</td>
<td>SC 289 – Lara Schmidt</td>
<td>0802</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Wednesday</td>
<td>SC 189 – Sebastian Lee</td>
<td>0901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Wednesday</td>
<td>SC 289 – Alyssa Welles</td>
<td>0902</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Wednesday</td>
<td>SC 189 – Virgil Hsieh</td>
<td>1001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Wednesday</td>
<td>SC 289 – Kuan-Po Chen</td>
<td>1002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Thursday</td>
<td>SC 189 – Nicki Schrank</td>
<td>1101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Thursday</td>
<td>SC 289 – Amirul Warsanah</td>
<td>1102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30</td>
<td>Thursday</td>
<td>SC 189 – Nicki Schrank</td>
<td>1201</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Thursday</td>
<td>SC 189 – Kyle Egbert</td>
<td>1301</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Thursday</td>
<td>SC 189 – Zaid Alami</td>
<td>1401</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Thursday</td>
<td>SC 289 – Jacob Harvey</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Thursday</td>
<td>SC 189 – John Grabarczyk</td>
<td>1501</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Friday</td>
<td>SC 189 – Pavan Saripalli</td>
<td>1601</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30</td>
<td>Friday</td>
<td>SC 289 – Jonathan Mulvaine</td>
<td>1602</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30</td>
<td>Friday</td>
<td>SC 189 – Pavan Saripalli</td>
<td>1701</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30</td>
<td>Friday</td>
<td>SC 289 – Chris Cadwallader</td>
<td>1702</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Friday</td>
<td>SC 189 – Upsham Dawra</td>
<td>1801</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>Friday</td>
<td>SC 289 – Chris Abernathy</td>
<td>1802</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td>Friday</td>
<td>SC 189 – Sebastian Lee</td>
<td>1901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Friday</td>
<td>SC 189 – Gloria Budiman</td>
<td>2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30</td>
<td>Friday</td>
<td>SC 289 – Matthew Lane</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>