Exam Rules/Notes:

1. Computer your name, your student ID#, and your FOUR DIGIT lab section on the answer sheet (write in AND bubble-in all requested information). **A list of lab sections is on the back of this exam.**
2. Make sure you have all 50 questions to this exam. You should answer every question. All questions only have one best answer. Please be careful when filling in your answers, any answer deemed unreadable by Instructional Data Processing will be considered incorrect. **Only the answer on your answer sheet can be considered when grading.**
3. **No questions will be answered during the exam.** We do not answer questions during the exam due to the limited number of staff member present. It is easier for us to “throw out” a test question with an error than it is to answer questions or to make an announcement. Manage your time accordingly.
4. You must assume that all C code is implemented on the sage.ics machine and would be compiled with the gcc compiler as set up during the first week of the semester. Some problem provide only a code segment to save space and such problems should not be interpreted as automatically failing to compile. **MATLAB is only relevant to problems 8 – 12 of this exam.**
5. **Protect your work and keep your focus on your own exam.** It is dishonest if you copy from another student or permit another student to copy from you. Anyone found to violate course policies will receive a failing grade for the course and will be referred to the Office of the Dean of Students.
6. **You are not permitted to use any resources during this exam.** This includes but is not limited to; texts, notes, calculators, cell phones, MP3 players, and computers. If your cell phone rings during the exam you will be required to immediately submit your exam and leave the testing facility.
7. When you are finished with the exam, please proceed to the FRONT to **submit your answer form AND exam before exiting the facility.** You will be **required to show photo identification** before your work can be accepted. You may not return to your seat once you have submitted your work.
8. **When time is called all writing must stop.** You are not permitted to continue to make revisions to any answer on your exam once time has been called.
9. **Your final exam score will appear on Blackboard by Saturday afternoon.** Do not contact course staff member about mistakes (if any) until answers are posted. If you feel an error remains after answers have been posted you make make an e-mail request (wcrum AT purdue DOT edu) from your Purdue e-mail account for a regrade by Tuesday December 22, 2009 (11:00am). Your request must include a description (using code if applicable) demonstrating your position as to why the posted answer is incorrect.
10. Final course grades and the announcement of the final cutoffs will be posted on the afternoon of Tuesday December 22, 2009. **Grades are ONLY assigned based on points earned.** Do not e-mail course staff members regarding cutoffs until AFTER they have been posted. Teaching assistants have no ability to revise grades after they assign an initial grade for a given assignment.
   - Information on how to contact course staff members once final course grades have been determined will be posted on Blackboard. Please allow sufficient time for your circumstance to be considered and a reply given. Failing to follow the specified instructions will only delay our ability to address your concern and may result in your request being denied without further consideration.
   - Regrade requests or Blackboard errors on older assignments must be addressed before leaving computer by printing up your request and sliding it under Bill Crum's office door (HAAS G-26).
1. Which of the following statements are false?
   A. The preprocessor prepares your program for compilation.
   B. The local declarations and executable statements of a function must never overlap.
   C. A failed attempt to compile a C program will result in any existing executable file being deleted.
   D. None of the Above.

2. What is the output of the first print statement?
   A. x = 0
   B. x = 1
   C. x = 2
   D. None of the Above

3. What is the output of the second print statement?
   A. y = 8
   B. y = 7
   C. y = 2
   D. None of the Above

4. Given the integers x = 0, y = 1, and z = -1, which of the following are false?
   A. !x && y && z
   B. x || y && z
   C. x++ && --y || ++z
   D. More than one of the Above

5. The image above represents the conversion rank of the various data types available in the C programming language. Which of the following methods of type conversions will always convert a lower ranking data type to a higher ranking data type when evaluating an expression that involves two or more data types?
   A. Implicit Type Conversions
   B. Explicit Type Conversions
   C. Both A and B
   D. None of the Above
Use the program below for problems 6 and 7

```c
#include<stdio.h>

#define MINGPA 3.50
#define FEES 2500
#define AWARD 1000

int main()
{
   int points;
   int hours;
   float gpa;
   float fee;

   printf("Enter the number of credit hours and points: ");
   scanf("%d %d", &hours, &points);

   fee = FEES - (int) (gpa / MINGPA) * AWARD;

   printf("fee = $ %.2f\n", fee);
   return(0);
}
```

6. Which of the following would correctly complete the calculation of the GPA on line 17 in the program above?

A. gpa = (float) (points / hours);
B. gpa = (float) points / hours;
C. gpa = 0.0 + points / hours;
D. None of the Above

7. What is the output of the program above if the user were to enter 9 and 33 as input (and given problem #6 was answered correctly)?

A. fee = $ 2500.00
B. fee = $ 1500.00
C. fee = $ 1425.38
D. None of the Above

NOTICE! Problems 8 – 12 are MATLAB questions! The rest of the exam is related to the C Language!

Use the MATLAB CODE below for problems 8 and 9

```matlab
n = 519324;
x = 0;
sum = 0;

while(n > 0)
   if(mod(n, 10) > x)
      x = mod(n, 10);
   else
      sum = sum + 1;
   end
   n = floor(n / 10);
end

fprintf('x = %d\n', x);
fprintf('sum = %d\n', sum);
```

8. What is the output of the first print statement?

A. x = 4
B. x = 5
C. x = 9
D. None of the Above

9. What is the output of the second print statement?

A. sum = 4
B. sum = 2
C. sum = 3
D. None of the Above
Use the MATLAB CODE below for problems 10 and 11

```matlab
base08 = 72;
octal = 0;
count = 0;

while(base08 > 0)
octal = octal + (mod(base08, 8) * 10 ^ count);
base08 = floor(base08 / 8);
count = count + 1;
end

fprintf('octal = %d
', octal);
fprintf('count = %d
', count);
```

10. What is the output of the first print statement?
   A. octal = 110
   B. octal = 48
   C. octal = 98
   D. None of the Above

11. What is the output of the second print statement?
   A. count = 1
   B. count = 2
   C. count = 3
   D. None of the Above

Use the MATLAB CODE below for problem 12

```matlab
x = 0;
y = 0;

for a = 5 : -1 : 1
    x = x + 1;
    for b = 1 : 2 : 5
        y = y + 1;
    end
end

fprintf('x = %d y = %d
', x, y);
```

12. What is the output of the first print statement?
   A. x = 4 y = 12
   B. x = 5 y = 10
   C. x = 5 y = 15
   D. None of the Above

NOTICE! Problems 8 – 12 are MATLAB questions! The rest of the exam is related to the C Language!
Use the code below for problems 13 – 15

```c
int addPlus2(int, int);
int subFrom2(int, int);

int main()
{
    int x = 2;
    int y = 7;
    int z = 0;

    z += addPlus2(x, y);
    printf("x + y = %d\n", x + y);
    printf("z = %d\n", z);
    x = 0;
    y = 0;
    z = subFrom2(x++, ++y);
    printf("x = %d y = %d z = %d\n", x, y, z);
    return(0);
}

int subFrom2(int y, int x)
{
    return(x / y - 2);
}

int addPlus2(int x, int y)
{
    x = x + y + x;
    y += 1;
    return(x);
}
```

13. What is the output of the first print statement in main?
   A. z = 11
   B. z = 13
   C. z = 16
   D. None of the Above

14. What is the output of the second print statement in main?
   A. x + y = 9
   B. x + y = 21
   C. x + y = 19
   D. None of the Above

15. What is the output of the third print statement in main?
   A. x = 1 y = 1 z = -2
   B. x = 0 y = 1 z = -2
   C. x = 1 y = 1 z = -1
   D. None of the Above
Use the code below for problems 16 – 17

```c
int makeCount(int*);

int main()
{
    int x = 21734;
    int y;

    y = makeCount(&x);
    printf("x = %d\n", x);
    printf("y = %d\n", y);
    return(0);
}

int makeCount(int *x)
{
    int ct = 0;
    int temp = *x;

    do
    {
        temp = *x % 100;
        if(temp % 3 == 0)
        {
            ct++;
        }
        *x /= 10;
    }while(*x > 0);

    return(ct);
}
```

16. What is the output of the first print statement in main?
   A. x = 0
   B. x = 2
   C. x = 21734
   D. None of the Above

17. What is the output of the second print statement in main?
   A. y = 2
   B. y = 1
   C. y = 0
   D. None of the Above

18. Which of the following statements is false?
   A. The control expression that follows the keyword switch must be an integral type which also includes the character data type.
   B. Each case label of a switch construct includes the keyword case followed by a constant expression.
   C. Multiple case labels of a switch construct can utilize the same symbolic constant in its constant expression as long as no two expressions result in the same value.
   D. None of the Above
Use the code segment below for problems 19 – 21

```c
int main()
{
    FILE *fptr;
    int max;
    int status;

    fptr = openFile();

    if(fptr)
    {
        do
        {
            max = getData(fptr, &status);
            if(status != EOF)
            { 
                displayProducts(max);
            }
        }while(status != EOF);
    }

    return(0);
}
```

19. Which of the following is the prototype of the `openFile` function given the `main` function on the left?
   A. `void openFile(FILE*)`;
   B. `FILE* openFile();`
   C. `FILE* openFile(FILE*)`;
   D. None of the Above

20. Which of the following is the prototype of the `getData` function given the `main` function on the left?
   A. `void getData(FILE*, int*)`;
   B. `int getData(FILE*, int*)`;
   C. `int* getData(FILE*, int);`
   D. None of the Above

21. Which of the following can replace the logical expression `if(fptr)` and produce the same results?
   A. `if(fptr == 0)`
   B. `if(fptr != 0)`
   C. `if(!fptr)`
   D. None of the Above

22. Which of the following library functions will return the EOF value?
   A. `fscanf`
   B. `fclose`
   C. `fopen`
   D. None of the Above

23. Which of the following will append the results of the executable file to another file named output?
   A. `a.out < data > output`
   B. `a.out < data >> output`
   C. `a.out < output > data`
   D. None of the Above

Use the code below for problems 24 and 25

```c
int calcCard(int);

int main()
{
    int x = 29;
    printf("Function Result = %d\n", calcCard(x));

    return(0);
}

int calcCard(int x)
{
    int sum = 0;

    if(x > 0)
    {
        sum = sum + x % 2 + calcCard(x / 2);
    }

    return(sum);
}
```

24. What is the output of the program on the left?
   A. Function Result = 1
   B. Function Result = 4
   C. Function Result = 3
   D. None of the Above

25. What is the total number of times that the `calcCard` function is called?
   A. 4
   B. 5
   C. 6
   D. None of the Above
26. Which of the following is the complement to the logical expression below?

\[ x > y \land \neg(x > 0 \land y > 0) \]

A. \[ x \leq y \lor (x < 0 \land y < 0) \]  
B. \[ x \leq y \lor (x > 0 \lor y > 0) \]  
C. \[ x \leq y \lor (x > 0 \land y > 0) \]  
D. None of the Above

Use the code segment below for problem 27

```c
switch(x % y) {
    case 1:
    case 0:  z++;  
}
```

27. Which of the following if statements is equivalent to the switch construct given above?

A. \( \text{if}(x \% y == 0 \lor 1) \)  
B. \( \text{if}(x \% y) \)  
C. \( \text{if}(x \% y == 0 \land 1) \)  
D. None of the Above

28. Which of the following describes the value returned from the \textit{strlen} function as found in the string.h library?

A. The declared size of the array.  
B. The total number of characters in the array up to and including the delimiter character.  
C. The index of the delimiter character in the array.  
D. None of the Above

29. Which of the following is true regarding the \textit{strcmp} and \textit{strcpy} functions as found in the string.h library?

A. \textit{strcmp} returns the sum of the differences between the two arrays being compared.  
B. The second parameter to the \textit{strcpy} function is the source string and the first parameter is the string to which the source is to be copied.  
C. \textit{strcmp} returns a value of 1 if the two strings being compared are equivalent.  
D. None of the Above

Use the program below for problems 30 – 32

```c
#include<stdio.h>
#define SIZE 50

int main()
{  
    int data[SIZE] = {1, 3, 5, 7};  
    char name[SIZE] = "Final Exam";

    printf("data = %d\n", data);
    printf("data[4] = %d\n", data[4]);
    printf("name[5] = %d\n", name[5]);

    return(0);
}
```

30. Which of the following is the output of the first print statement?

A. data = 1  
B. data = 1357  
C. data = 50  
D. None of the Above

31. Which of the following is the output of the second print statement?

A. data[4] = 0  
B. data[4] = 7  
C. data[4] = 9  
D. None of the Above

32. Which of the following is the output of the third print statement?

A. name[5] =  
B. name[5] = 32  
C. name[5] = 108  
D. None of the Above
33. Given the array below and the array after two passes through one of the sorting algorithms, which of the following sorting algorithms has been used?

Original Array: 7 5 9 3 1 6

Array After Two Passes: 7 5 9 1 3 6

A. Selection
B. Insertion
C. Bubble
D. More than one of the Above

34. Given the array below and the array after two passes through one of the sorting algorithms, which of the following sorting algorithms has been used?

Original Array: 7 5 9 3 1 6

Array After Two Passes: 7 5 9 6 3 1

A. Selection
B. Insertion
C. Bubble
D. More than one of the Above

35. Given the array below and the array after two passes through one of the sorting algorithms, which of the following sorting algorithms has been used?

Original Array: 7 5 9 3 1 6

Array After Two Passes: 9 7 5 6 3 1

A. Selection
B. Insertion
C. Bubble
D. More than one of the Above

Use the function below for problem 36

```c
void sort(int y[], int n)
{
    int lcv;
    int temp;
    int numPasses;

    for(numPasses = 1; numPasses <= n; numPasses++)
    {
        temp = y[numPasses];
        lcv = numPasses - 1;

        while(temp < y[lcv] && lcv >= 0)
        {
            y[lcv + 1] = y[lcv];
            lcv--;
        }

        y[lcv + 1] = temp;
    }
}
```

36. Which of the following sorting algorithms has been implemented in the function provided?
A. Selection
B. Insertion
C. Bubble
D. None of the Above
Use the function below for problem 37

```c
int getData(int d[]) {
    int lcv = 0;
    do {
        printf("Enter integer #\%2d or %d to EXIT: ", lcv + 1, EXIT);
        scanf("%d", &d[lcv]);
        lcv++;
    } while (d[lcv - 1] != EXIT && lcv < LIMIT);
    //MISSING RETURN STATEMENT TO BE PLACED HERE
}
```

37. Which of the following return statements will complete the function above such that the number of valid integers entered will be returned? The EXIT value is not considered to be part of the data set. You may assume that LIMIT is the declared size of the array.

A. `return(d[lcv - 1] == EXIT ? lcv - 1 : lcv);`
B. `return(lcv < LIMIT ? lcv - 1 : lcv);`
C. `return(d[lcv] == EXIT ? lcv - 1 : lcv);`
D. None of the Above

38. Given the array below, which of the following represents the values of the variable mid in the binary searching algorithm in the order it searches array for the designated target value?

```c
int x[9] = {8, 10, 11, 13, 17, 19, 21, 25, 27};
int target = 25;
```

A. 4, 6, 7
B. 4, 7
C. 5, 7
D. None of the Above

39. Given the array below, which of the following represents the values of the variable mid in the binary searching algorithm in the order it searches array for the designated target value?

```c
int x[9] = {8, 10, 11, 13, 17, 19, 21, 25, 27};
int target = 12;
```

A. 4, 1, 2, 3
B. 4, 2, 3
C. 4, 2
D. None of the Above
Use the program below for problems 40 and 41

```c
#include<stdio.h>
define SIZE 5

void makeCounts(int[], int);

int main()
{
  int ct[SIZE] = {0, 0, 0, 0, 0};
  int lcv;

  makeCounts(ct, 482610);

  for(lcv = 0; lcv < SIZE; lcv++)
  {
    printf("%d", ct[lcv]);
  }

  printf("\n");

  return(0);
}

void makeCounts(int ct[], int x)
{
  while(x > 10)
  {
    printf("%d\n", x % 100);
    ct[(x % 100) % 5]++;
    x /= 10;
  }
}
```

40. Which of the following values is NOT displayed among the output generated in the `makeCounts` function?
   A. 10
   B. 26
   C. 48
   D. None of the Above

41. Which of the following is the output generated in the `main` function?
   A. 12110
   B. 01121
   C. 01211
   D. None of the Above
Use the program below for problems 42 – 44

```c
#include<stdio.h>

void mod3(int);
void mod2(int[], int, int);
void mod4(int*);

int main()
{
    int x[4] = {5, 10, 15, 20};
    mod3(x[2]);
    printf("x[2] = %d\n", x[2]);
    mod4(&x[3]);
    printf("x[3] = %d\n", x[3]);
    mod2(x, 0, 1);
    printf("x[0] = %d x[1] = %d\n", x[0], x[1]);
    return(0);
}

void mod2(int x[], int s, int e)
{
    while(s <= e)
    {
        x[s] = x[s] % 2;
        s++;
    }
}

void mod4(int *x)
{
    *x = *x % 4;
}

void mod3(int x)
{
    x = x % 3;
}
```

42. Which of the following is the output of the first print statement?
   A. x[2] = 3
   B. x[2] = 15
   C. x[2] = 0
   D. None of the Above

43. Which of the following is the output of the second print statement?
   A. x[3] = 4
   B. x[3] = 20
   C. x[3] = 0
   D. None of the Above

44. Which of the following is the output of the third print statement?
   A. x[0] = 5 x[1] = 10
   B. x[0] = 1 x[1] = 0
   C. x[0] = 0 x[1] = 1
   D. None of the Above

45. Given an array of a defined SIZE, how many passes through the bubble sorting algorithm are required to guarantee that they array will be sorted?
   A. SIZE
   B. SIZE – 1
   C. SIZE + 1
   D. None of the Above

46. Given an array of a defined SIZE, on which pass through the selection sorting algorithm are two values moved from the unsorted list to the sorted list?
   A. The first pass
   B. The SIZE pass
   C. The SIZE – 1 pass
   D. None of the Above
Use the program below for problem 47

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

int getAmtData();
void getData(int[], int);
void printData(int[], int);

int main()
{
    int *x;
    int numData;
    numData = getAmtData();
    getData(x, numData);
    printData(x, numData);
    return(0);
}
```

47. Which of the following would correctly allocate enough memory to accommodate `numData` integers in memory and assign the pointer variable `x` to refer to that memory? The resulting expression would be placed on line 15 in the code above.
   A. `x = (int*)malloc(sizeof(int) + numData);`
   B. `x = (int*)malloc(sizeof(numData));`
   C. `x = (int*)malloc(sizeof(int) * numData);`
   D. None of the Above

48. Which of the following will never be a result of exceeding the bounds (defined size) when manipulating elements of an array?
   A. No problem, the program will work as expected.
   B. The program executes, but the results are unexpected.
   C. The program crashes.
   D. None of the Above

49. How many integers can the multidimensional array defined below store in memory?

   ```
   int coordinates[10][5][3];
   ```

   A. 18
   B. 150
   C. 45
   D. None of the Above

50. Where will you be able to find any end of semester announcements including your final exam score and the release of final cutoffs for the semester?
   A. Blackboard
   B. By e-mailing or calling the staff members of the course.
   C. Asking the guy down the hall that you think might be in the class or at least you think he knew you were taking this class.
   D. The answer is A. Select any other answer and you won't get points for this problem!
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></td>
</tr>
<tr>
<td>! ~</td>
<td>Logical negation/bitwise complement</td>
<td></td>
</tr>
<tr>
<td>(type)</td>
<td>Cast (change type)</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Dereference</td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>sizeof</td>
<td>Determine size in bytes</td>
<td></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, Bitwise shift right</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&lt; &lt;=</td>
<td>Relational less than/less than or equal to</td>
<td>left-to-right</td>
</tr>
<tr>
<td>&gt; &gt;=</td>
<td>Relational greater 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></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></td>
</tr>
<tr>
<td>%= &amp;</td>
<td>Modulus/bitwise AND assignment</td>
<td></td>
</tr>
<tr>
<td>^=</td>
<td>=</td>
<td>Bitwise exclusive/inclusive OR assignment</td>
</tr>
<tr>
<td>&lt;&lt;= &gt;&gt;&gt;=</td>
<td>Bitwise shift left/right assignment</td>
<td></td>
</tr>
<tr>
<td>,</td>
<td>Comma (separate expressions)</td>
<td>left-to-right</td>
</tr>
<tr>
<td>Char</td>
<td>Dec</td>
<td>Char</td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>(nul)</td>
<td>0</td>
<td>space</td>
</tr>
<tr>
<td>(soh)</td>
<td>1</td>
<td>!</td>
</tr>
<tr>
<td>(stx)</td>
<td>2</td>
<td>&quot;</td>
</tr>
<tr>
<td>(etx)</td>
<td>3</td>
<td>#</td>
</tr>
<tr>
<td>(eot)</td>
<td>4</td>
<td>$</td>
</tr>
<tr>
<td>(enq)</td>
<td>5</td>
<td>%</td>
</tr>
<tr>
<td>(ack)</td>
<td>6</td>
<td>&amp;</td>
</tr>
<tr>
<td>(bel)</td>
<td>7</td>
<td>39</td>
</tr>
<tr>
<td>(bs)</td>
<td>8</td>
<td>(</td>
</tr>
<tr>
<td>(ht)</td>
<td>9</td>
<td>)</td>
</tr>
<tr>
<td>(nl)</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>(v)</td>
<td>11</td>
<td>+</td>
</tr>
<tr>
<td>(np)</td>
<td>12</td>
<td>,</td>
</tr>
<tr>
<td>(cr)</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>(so)</td>
<td>14</td>
<td>.</td>
</tr>
<tr>
<td>(si)</td>
<td>15</td>
<td>/</td>
</tr>
<tr>
<td>(dle)</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>(dc1)</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>(dc2)</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>(dc3)</td>
<td>19</td>
<td>3</td>
</tr>
<tr>
<td>(dc4)</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>(nak)</td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>(syn)</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>(etb)</td>
<td>23</td>
<td>7</td>
</tr>
<tr>
<td>(can)</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>(em)</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>(sub)</td>
<td>26</td>
<td>:</td>
</tr>
<tr>
<td>(esc)</td>
<td>27</td>
<td>;</td>
</tr>
<tr>
<td>(fs)</td>
<td>28</td>
<td>&lt;</td>
</tr>
<tr>
<td>(gs)</td>
<td>29</td>
<td>=</td>
</tr>
<tr>
<td>(rs)</td>
<td>30</td>
<td>&gt;</td>
</tr>
<tr>
<td>(us)</td>
<td>31</td>
<td>?</td>
</tr>
<tr>
<td>Lab Instructor</td>
<td>Lab Day / Time</td>
<td>Section</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>---------</td>
</tr>
<tr>
<td>Alyssa Wells</td>
<td>Wednesday – 7:30</td>
<td>0101</td>
</tr>
<tr>
<td>Jessica Sparks</td>
<td>Wednesday – 11:30</td>
<td>0201</td>
</tr>
<tr>
<td>Sebastian Lee</td>
<td>Wednesday – 1:30</td>
<td>0301</td>
</tr>
<tr>
<td>Vibhore Singh</td>
<td>Wednesday – 3:30</td>
<td>0401</td>
</tr>
<tr>
<td>Jessica Sparks</td>
<td>Thursday – 7:30</td>
<td>0501</td>
</tr>
<tr>
<td>Mike James</td>
<td>Thursday – 9:30</td>
<td>0601</td>
</tr>
<tr>
<td>Kuan-Po Chen</td>
<td>Thursday – 11:30</td>
<td>0701</td>
</tr>
<tr>
<td>Fan Zhang</td>
<td>Thursday – 1:30</td>
<td>0801</td>
</tr>
<tr>
<td>Fan Zhang</td>
<td>Thursday – 3:30</td>
<td>0901</td>
</tr>
<tr>
<td>Aaron Lentner</td>
<td>Friday – 7:30</td>
<td>1001</td>
</tr>
<tr>
<td>Vibhore Singh</td>
<td>Friday – 11:30</td>
<td>1101</td>
</tr>
<tr>
<td>Sebastian Lee</td>
<td>Friday – 1:30</td>
<td>1201</td>
</tr>
</tbody>
</table>