Time taken (before correction and edits): 1 : 09

Question 1: HTML and CGI

```
<html>
  <header><title>Login</title></header>
  <body bgcolor="blue"> <!--Background color-->
    <p font="Times New Roman" size="23"> WELCOME </p>
    <table>
      <tr> <!-- Row 1 -->
        <th><h2>LOGIN</h2></th>
        <th><h2 color="red">REGISTER</h2></th>
      </tr>
      <tr> <!-- Row 2 -->
        <td><form action="login.exe" method="POST">
            USER NAME: <input type="text" name="un">
            <br>
            PASSWORD: <input type="password" name="pw">
            <br>
            <input type="submit" value="LOGIN">
        </td>
        <td><form action="register.py" method="POST">
            REAL NAME: <input type="text" name="rn">
            <br>
            USER NAME: <input type="text" name="un">
            <br>
            PASSWORD: <input type="password" name="pw">
            <br>
            EMAIL: <input type="text" name="em">
            <br>
            MALE: <input type="radio" name="gend" value="M">
            FEMALE: <input type="radio" name="gend" value="F">
            <br>
            <input type="submit" value="REGISTER">
          </form>
        </td>
      </tr>
```

```
    </table>
    <p font="Ariel"><center><a href="help.htm">
        Click Here For Help</a><center></p>
  </body>
</html>
```

Question 2: GNU

(A)

(a)

```
CC = gcc # Sets a variable to gcc, for compiler
CFLAGS = -g -Wall # Var for compiler flags

count: countwords.o counter.o scanner.o
# If the 3 o files exist, make the count file
# Or if one of the o files is newer than the count file
    $(CC) $(CFLAGS) -o count countwords.o counter.o scanner.o
    # Compiles all the o files as count, with all the warnings
    # and inserts timestops for GProf

countwords.o: countwords.c scanner.h counter.h
# Make countwords.o if source and corresponding headers
# are present
    $(CC) $(CFLAGS) -c countwords.c
    # Makes countwords.o, displays all errors, GProf

counter.o: counter.c counter.h
# Make counter.o if source and headers are there
    $(CC) $(CFLAGS) -c counter.c
    # counter.o, same as above

scanner.o: scanner.c scanner.h
    $(CC) $(CFLAGS) -c scanner.c
    # Same as counter.o

clean:
```

```
$(RM) count *.o *~
# Removes count and all .o files and files ending in ~
```

(b) Scanner.o depends on scanner.h and scanner.c, counter.o depends on counter.h and counter.c, countwords.o depends on countwords.c scanner.h counter.h and count depends on all .o files.

(c) Huge box that is just the count program. Within it are smaller rectangles, one for countwords.o, one for counter.o and one for scanner.o

(B)

```
(gdb) break main
(gdb) run
(gdb) step
(gdb) step
(gdb) ... # Until in makeit()
(gdb) step
(gdb) step
(gdb) ... # Eventually program will crash
(gdb) where # Find out where you are when it crashed
(gdb) backtrace
```

Question 3: C Programming

```
// Function header, returns int, takes int pointer
int foo(int *pn)
{
    int c, s; // Makes 2 ints
    // Loops while user inputs a space, CR or tab
    while ((c=getch() == ' ' || c == '\n' || c == '\t'));

    s = 1;
    if (c == '+' || c == '-') // c is + or -
    {
    // If c is +, s=1, else s=-1
    s = (c=='+') ? 1 : -1;
    c = getch(); // Get a char again
    }
```

```
    // Make contents of pointer 0
    // Loop while c is inbetween 0 and 9 (ASCII)
    // Get character at end of each loop
    for(*pn=0; c >= '0' && c <= '9'; c=getch())
        // 10 times pointers val + # that c stores
        *pn = 10 * *pn + c      '0';

    *pn *= s; // Make pn the same sign as s
    if (c != EOF) ungetch(c); // If not end of file
    return c; // return 2nd to last char obtained
}
```

This program gets a sign from the user, then keeps getting numbers from the user, summing them up and multiplying by 10 everytime a new number comes along. This will be available to the user since it's a pointer, but the actual function returns the 2nd to last obtained char.

Question 4: C Programming

```
#include <stdio.h>
#include <string.h>
struct STUDENT{
  char name[50];
  double gpa;
};

int main(int argc, char* argv[]){
  int n; // Input of user
  printf("How many students in the classroom?\n");
  scanf("%d",&n); // Scan number into n
  struct STUDENT p*; // Pointer for array of students
  // Malloc for n students
  p = (struct STUDENT *)malloc(n*(sizeof(STUDENT)));
  int j = 0; // Counter for how many entries csv has

  FILE* in = fopen("GPA.CSV","rt"); // Open file
  if(in==NULL){ //File doesn't exist
    printf("GPA.CSV does not exist! \n");
```

```
  }
  else{
    // For storing values
    char name[50];
    double gpa;
    // Scan until EOF
    while(scanf("%s,%lf",name,&gpa)!=EOF){
      j++; // Increment amount of entries
      if(j<n){ // Did not hit max yet
        // p+j-1 since j is one bigger than index
        strcpy((p+j-1)->name,name); // copy name
        (p+j-1)->gpa = gpa; // Copy gpa
      }
    }
    fclose(in); // Close input
  }
  // Loop to get average, j entries
  double average = 0;
  for(int i=0; i<j; i++){
    average += (p+i)->gpa; // Sum gpas
  }
  average /= j; // Divide by number of entries
  printf("Average GPA: %lf",average);
}
```

Question 5: Python Programming

```
import cgi

input = cgi.FieldStorage() # Get input

# Get the fields we want
filename = input["filename"]
lines = int(input["lines"]) # Cast to int

# HTML output
print("Content-type: text/html\n\n")
```

```
print("<html>")
print("<header><title>Products</title></header>")
print("<body>")
# Try opening CSV file
try:
    file = fopen(filename,"r")
except: # Error
    print("File not found")
else:
    print("<table>")
    print("<tr>")
    print("<th> Product Code </th>")
    print("<th> Quantity in Stock</th")
    print("<th> Min in Stock</th>")
    print("<th>Unit Price</th>")
    print("<th> Min to Purchase</th>")
    for i in range(lines): # Loop through lines
        data = file.readline()
        # Tokenize?
        code = ""
        quantity = ""
        min = ""
        unit = ""
        min2 = min-quantity
        if quantity<min:
            print("<tr>")
            print("<td> %d </td>"%(data))
            print("<td> %d </td>"%(quantity))
            print("<td> %d </td>"%(min))
            print("<td> %d </td>"%(unit))
            print("<td> %d </td>"%(min2))
            print("</tr>")

    print("</table>")
print("</body>")
print("</html>")
```