Problem 1:

ABC Trucking has asked you to create a program that sends out exactly the number of trucks needed to pick up orders from different potato farms. Each potato farm lists the number of potatoes available from each part of the farm in a file with the same name as the potato farm (see sample run). Note that each truck for ABC trucking can take 3 boxes and each box holds 150 potatoes.

```
(base) Computers-MacBook-Air:C computer$ gcc -o potato practice.c
(base) Computers-MacBook-Air:C computer$ ./potato
1. Accept orders
2. Exit
1
Enter company name: Dans
File didn't open...can't send trucks.

1. Accept orders
2. Exit
1
Enter company name: Dans PoTaToes
Opening file: dans_potatoes.txt
   ***Total trucks to send: 5

1. Accept orders
2. Exit
1
Enter company name: Ultans Potatoes
Opening file: ultans_potatoes.txt
   ***Total trucks to send: 3

1. Accept orders
2. Exit
2
```

a negative number means number of non-sellable potatoes. We will ignore a line with a negative number.

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

void space_to_underscore(char line[], int length);
int total_trucks(char filename[]);


int main (int argc, char **argv)
{
  char line[40];

  while(1) /*I don't like running a loop this way, but just showing you…I will be forced to use break to exit*/
```

```
  {
    printf("1. Accept orders\n2. Exit\n");
    fgets(line,40,stdin);
    strtok(line,"\n");

    if(!strcmp(line,"1") || !strcmp(line,"Accept orders"))
    {
      printf("Enter company name: ");
      fgets(line,40,stdin);
      strtok(line,"\n");

      space_to_underscore(line, strlen(line));
      strcat(line,".txt"); /*create the whole file*/

      int num=total_trucks(line);

      if(num==-1)
      {
        printf("File didn't open...can't send trucks.\n\n");
      }

      else
      {
        printf("   ***Total trucks to send: %d\n\n",num);
      }

    }

    else if(!strcmp(line,"2")|| !strcmp(line,"exit"))
    {
      break;
    }

    else
    {
      printf("Unknown input.\n");
    }

  }

}



/*this function turns spaces to underscores and makes sure all letters are lowercase*/
void space_to_underscore(char line[], int length)
{
  int i;

  for(i=0;i<length;i++)
```

```
  {
    if(line[i]==' ')
    {
      line[i]='_';
    }

    else if(65<=line[i] && line[i]<=90) /*Using ascii to check for any uppercase*/
    {
      line[i]+=32;
    }

    /*no final else means do nothing in any other case*/
  }
}


int total_trucks(char filename[])
{
  FILE* fp=fopen(filename,"r");
  char line[40];
  int total_potatoes=0, current;

  if(!fp)
  {
    return -1;
  }

  else
  {
    printf("Opening file: %s\n",filename);

    while(fgets(line,40,fp))
    {
      current=atoi(line);
      if(0<=current)
      {
        total_potatoes+=current;
      }
    }

    /*get total trucks-150 potatoes a box, 3 boxes a truck*/
    int total_boxes=total_potatoes/150;

    if(total_potatoes%150) /*check for leftover potatoes*/
    {
      total_boxes++;
    }

    int total_trucks=total_boxes/3;
    if(total_boxes%3)
```

```
    {
      total_trucks++;
    }
    return total_trucks;
  }
}

/*How it works example:
        total_potatoes=800
        800/150=5 (meaning we have 5 boxes full of 150 potatoes each)
        800%150=50 (meaning we have 50 leftover potatoes…so 800 divided by 150 gives 5 remainder 50)
        we have to now add 1 more box (to hold these 50 leftover potatoes…it won't be full)…so total boxes is 5 (full boxes)+1 (not
        full)=6

        Trucks (same concept as above):
        6/3=2
        6%3=0 (no leftover boxes, so only need the 2 trucks…if it had been 7 boxes for example, then 7%3==1, so we would need 2+1
        trucks (to get that leftover box)*/
```