Student Name	
Student ID	

Question 1: Provide a short answer for each of the following terms. Your answer must fit within the space provided in the table. [20 points, 2 points for each definition] – see crib sheet at end.

QUESTION	ANSWER
What number does sizeof(int) return?	Older computers 2 Newer computers 4
What C command can be used to terminate a C program early. Full syntax.	void exit(int status);
What does #ifndef mean?	It is a conditional test that resolves to TRUE only when the label, as in #IFNDEF LABEL, was NOT previously defined, as in #DEFINE LABEL, otherwise it resolves to TRUE.

Write a makefile that implements modular programming for this case: f1.c f1.h f2.c f2.h. f1.c includes f2.h and f2.c includes f1.h. Program name FF12.	`FF12: f1.o f2.o`
	`  gcc -o FF12 f1.o f2.o`
	`f1.0: f1.c f2.h`
	`  gcc -c f1.c`
	`f2.o: f2.c f1.h`
	`  gcc -c f2.c`

What does x receive? x=(5>10)?5:10;	10
Assume we have a function, fn, that has two int parameters. Write the function prototype that implements call-by-reference for fn.	void fn (int *x, int *y);
When is it better to use call-by-value over call-by -reference?	When you want to protect the value of the original variables.
In pointers, what does the asterix (*) and the ampersand (&) do?	Asterix: *var gives you access to the value pointed to by var Ampersand: &var gives you the address, as an integer, of var Eg: int x=10; int *p = &x; printf("%d", *p); // prints 10
Assume NODE *p points to a node with int data and NODE *next. Write the C statement to print the data field to the screen using p.	printf("%d", p->data);
What is the different between: **char *p = p + 1; and** **int *q = q + 1;**	The pointer p is incremented by 1 byte. The pointer q is incremented by 2 bytes (on older computer), 4 bytes on newer computers. (part marks for not specifying bytes)

Question 2: Write the following C program in the space provided [20 points]

Create a BANKACCOUNT struct with two fields: char name[100] and float balance. The account number is the index of the array. Create an array of 2 bank accounts. Initialize the array with: ABC having $100, and DEF having $200. Modify this array with data from a CSV file named transactions.csv. The format of the CSV file is as follows:

accountNumber,amount
accountNumber,amount

The file has many transactions. Read each record and then update the balance for the correct account, if it exists. The amount field is either a positive or negative number. After processing all the transactions from the file, display the balance from both accounts to the screen. Assume accountNumber and amount are valid. (you can use the back of the page for extra space)

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

struct BANK { char name[100]; float balance; } accounts[2];

int main() {
  FILE *p;
  char buffer[1000], token[100];
  int pos1, pos2, index;
  float amount;

  strcpy(accounts[0].name, "ABC"); accounts[0].balance = 100.0;
  strcpy(accounts[1].name, "DEF"); accounts[1].balance = 200.0;

  p = fopen("transactions.cvs", "rt");
  if (p == NULL) exit(1);

  fgets(buffer,999,p);
  while(!feof(p)) {
    token[0]=buffer[0]; token[1]='\0';
    index=atoi(token);

    for(pos1=2, pos2=0; buffer[pos1]!='\0'; pos1++, pos2++)
      token[pos2] = buffer[pos1]; // could have used \r \n in loop

    amount = atof(token);

    if (index<2 && index >= 0) accounts[index].balance += amount;
    fgets(buffer,999,p);
  }

  for(pos1=0; pos1<2; pos1++) printf("%.2f\n", accounts[pos1].balance);
  return 0;
}
```

CRIB SHEET

STRING.H
- Int strcmp(char *, char *);
- Int strlen(char *);
- Char *strcpy(char *, char *);
- Char *strcat(char *. char *);

STDIO.H
- int getchar();
- int puts(char *);
- int getc(FILE *);
- char *fgets(char *, int, FILE *);
- FILE *fopen(char *, char *);
- Int feof(FILE *);
- Void fclose(FILE *);
- Int fputc(int, file *);
- Int fputs(char *, FILE *);

STDLIB.H
- Int system(char *);
- Foat atof(char *);
- Int atoi(char *);
- Int abs(int);

MATH.H
- Double sqrt(double);
- Double pos(int, int);
- Int abs(int);
- Double floor(double);
- Double ceil(double);

CTYPE.H
- Int toupper(int);
- Int tolower(int);
- Int isalpha(int);
- Int isalphanum(int);
- Int isdigit(int);