

CS 100 Final Exam – Coding – Spring 2017 – Version A

- Create a directory called **final** and move into that directory to complete the programs below.
- The skeleton code for each problem can be downloaded from the course Blackboard site.
- You are not allowed to use the Internet. You can use **cs-intro** to test your programs.

1. Name this program **one.c** – This program generates a linked list of numbers. After it generates the linked list of numbers, it calls **aboveAverage** which prints all numbers in the linked list that are “above average.” That is, the value is greater than the average of all the numbers in the linked list. Write **aboveAverage**.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node {
    int value;
    struct node *next;
} Node;
void aboveAverage(Node *);
int main( void ) {
    int num;
    Node *head = NULL;
    printf("Enter a number to add to the list : ");
    scanf("%d", &num);
    while ( num != -999 ) {
        Node *ptr = (Node *) malloc ( sizeof(Node) );
        ptr->value = num;
        ptr->next = head;
        head = ptr;
        printf("Enter another number or -999 to exit : ");
        scanf("%d", &num);
    }
    aboveAverage(head);
    return 0;
}
```

2. Name this program **two.c** – This program reads two strings and uses a function to determine whether or not the two strings are the same if you ignore case (it does not matter if a given character in the string is upper-case or lower-case). The function **ignoreCase** returns a 1 if the strings are identical (ignoring case) and returns 0 otherwise. Write **ignoreCase**.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int ignoreCase(char *, char *);
int main( void ) {
    char input1[1024], input2[1024];
    printf("Enter a string : ");
    scanf("%s", input1);
    printf("Enter another string : ");
    scanf("%s", input2);
    if ( ignoreCase(input1, input2) )
        printf("%s and %s are the same if you ignore case\n", input1, input2);
    else
        printf("%s and %s are not the same, even ignoring case\n",
            input1, input2);
    return 0;
}
```

3. Name this program **three.c** – This program allocates space for a one-dimensional array (size specified by the user) and initializes that array. It then determines whether or not the array alternates between odd and even (or even and odd) values. For example, the first two arrays below alternate, while the third does not. Recall that **0** is an even number. Write the three functions shown in red.

[1 4 3 8 15]

odd then even then odd then even then odd

[2 9 10]

even then odd then even

[2 6 9 10]

even then even, fails

- **allocateArray** allocates space for an array of “size” integers
- **readArray** reads in the array of values
- **alternates** returns 1 if the array “alternates” and returns 0 otherwise

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

int *allocateArray(int);
void readArray(int *, int);
int alternates(int *, int);

int main( void ) {
    int size;
    printf("Enter the size of the array : ");
    scanf("%d", &size);
    int *array = allocateArray(size);
    printf("Enter array values (%d values) : ", size);
    readArray(array, size);
    int ans = alternates(array, size);
    if (ans == 1)
        printf("The array alternates\n");
    else
        printf("The array does not alternate\n");
    return 0;
}
```

Submit your exam

Compress the **final** directory containing your programs into a single (compressed) file.
Submit that compressed file to Blackboard.