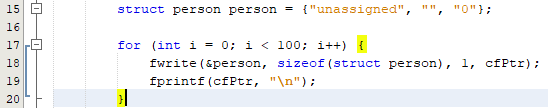
**Sean Poston**

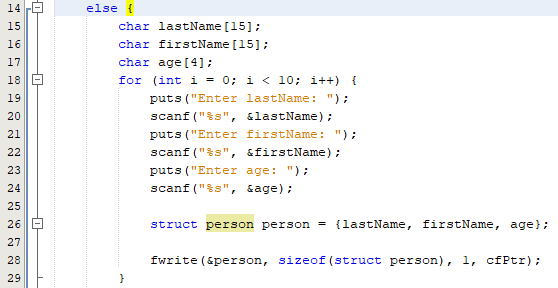
**CS351 Homework 8**

**11.11:**

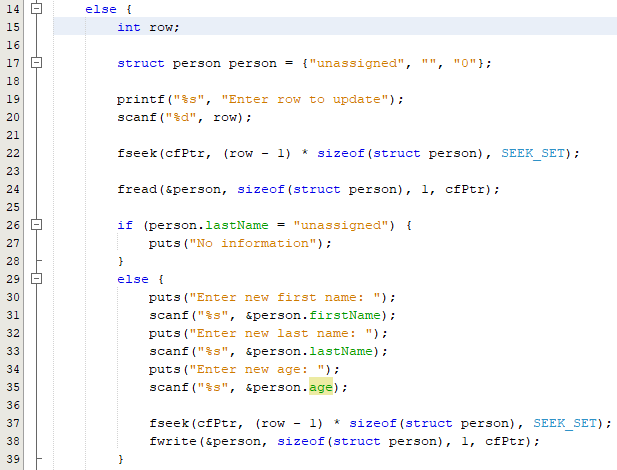
**a)**



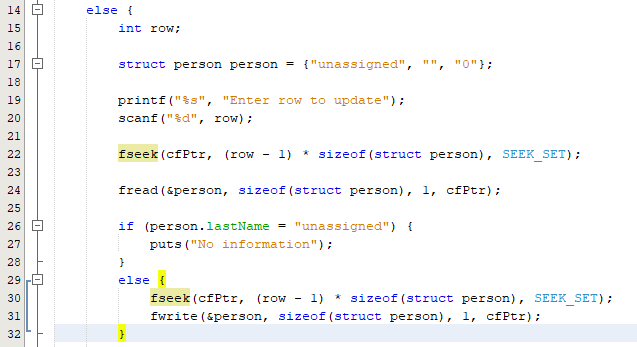
**b)**



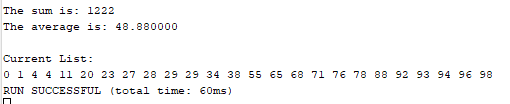
**c)**



**d)**



**12.8:** <https://repl.it/@seanposton4/advancedRandom>



**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

struct listNode {

char data;

struct listNode \*nextPtr;

};

typedef struct listNode ListNode;

typedef ListNode \*ListNodePtr;

void insertionSort(int \* list) {

int j = 1;

int temp;

for (int i = 1; i < 25; i++) {

j = i;

while (j > 0 && list[j - 1] < list[j]) {

temp = list[j - 1];

list[j - 1] = list[j];

list[j] = temp;

j--;

}

}

}

void insert(ListNodePtr \*sPtr, int value);

void sumList(ListNodePtr currentPtr);

void printList(ListNodePtr currentPtr);

int main(void) {

time\_t t;

srand(time(&t));

int list[25];

for(int i = 0; i < 25; i++)

list[i] = (rand() % 100);

insertionSort(list);

ListNodePtr startPtr = NULL;

for (int i = 0; i < 25; i++) {

insert(&startPtr, list[i]);

}

sumList(startPtr);

puts("");

printList(startPtr);

return 0;

}

void insert(ListNodePtr \*sPtr, int value) {

ListNodePtr newPtr;

ListNodePtr previousPtr;

ListNodePtr currentPtr;

newPtr = malloc(sizeof(ListNode));

if(newPtr != NULL) {

newPtr->data = value;

newPtr->nextPtr = NULL;

previousPtr = NULL;

currentPtr = \*sPtr;

while (currentPtr != NULL && value > currentPtr->data) {

previousPtr = currentPtr;

currentPtr = currentPtr->nextPtr;

}

if (previousPtr == NULL) {

newPtr->nextPtr = \*sPtr;

\*sPtr = newPtr;

}

else {

previousPtr->nextPtr = newPtr;

newPtr->nextPtr = currentPtr;

}

}

else {

puts("no.");

}

}

void sumList(ListNodePtr currentPtr) {

int sum = 0;

while(currentPtr != NULL) {

sum += currentPtr->data;

currentPtr = currentPtr->nextPtr;

}

printf("The sum is: %d\nThe average is: %f\n", sum, (float)sum / 25.0);

}

void printList(ListNodePtr currentPtr) {

puts("Current List:");

while (currentPtr != NULL) {

printf("%d ", currentPtr->data);

currentPtr = currentPtr->nextPtr;

}

}