Page 89-91 Excercises

#8

//Given an integer n > 0, write a recursive C++ function that returns the sum of the squares of 1 through n.

#include <iostream>

using namespace std;

int recSum(int n){

if (n==0){

return 0;

}

else{

return n+recSum(n-1);

}

}

int main(){

cout<<recSum(3)<<endl;

}

#13

The octal algorithm works by using a well known decimal to base-x algorithm. The decimal number is divided by the base and the remainder displayed until it no longer can be divided.

100/8 = 12 remainder 4 4

12/8= 1 remainder 4 4

1/8= 0 remainder 1 1

1\*64 + 4\*8 + 4\*1 = 64+32+4 = 100

100 in Octal is 144

#15

Pass by Value

6 2

7 1

8 0

8 0

7 1

6 2

When the recursive functions hits its break case, the function unwinds and repeats back the previous values. The values are exactly the same from its initial loop since the values from the previous loops are unchanged.

Pass by Reference

6 2

7 1

8 0

8 0

8 1

8 2

void recurse(int& x, int y)

When pass by reference, the x value is created in memory and changed during the recursive passes. Therefore after the break case is finished and the function begins to unwind, the x value is continually 8 since the value in the memory is being referenced.

#18

a.

int itPower(int x, int n){

int total = 1;

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

total \*=x;

}

return total;

}

b.

int recPower(int x, int n){

if (n==0){

return 1;

}

else{

return x\*recPower(x,n-1);

}

}

Page 156

#5

template<class ItemType>

LinkedBag<ItemType>::LinkedBag(const ItemType arr[], int size): headPtr(nullptr),itemCount(0){

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

bool j = add(arr[i]);

}

}

#11

Steps to add to a double linked list

1. Create new node with entry, add data, save to temporary pointer
2. Get headPtr, assign to the new node’s next, assign new node’s previous to null
3. Change previous of head node to the new node
4. Assign head ptr to new node’s address

#12

Remove first node in double linked list

1. get address of second node through head node’s next
2. save address to headPtr
3. delete address, assign to nullptr