**Program#2**

;Christopher McCoy, this program demonstrates preorder notation of equations.

( \* 100(expt (+ 1 .09) 20))

(- (\* (/ 100 .0075) (- (expt (+ 1 .0075) (+ 36 1)) 1) ) 100)

(/ (\* 5000 .015)(- 1 (expt (+ 1 .015) ( - 60 ))))

(/ (\* 2000 .005)(- (expt ( + 1 .005) 11) 1))

(\* (expt (+ 1 .01) ( - 4 )) 100000)

**Program#4**

;Christopher McCoy, this program is example of how to use cond and if statements in scheme.

( define (waterWeight diameter length pureWater)

(if pureWater

( \* 3.14159 (/ diameter 24) length 62.4)

( \* 3.14159 (/ diameter 24) length 64.08)

)

)

(define (convK MHz)

(\* MHz 1000)

)

(define (radio frequency)

(cond ((<= (convK frequency) 30)

"VLF")

((<= (convK frequency) 300)

"LF")

((<= frequency 3)

"MF")

((<= frequency 30)

"HF")

((<= frequency 328.6)

"VHF")

((<= frequency 2009)

"UHF")

)

)

(define (divide x)

(/ x 100)

)

(define (payCheck hourly quantity rate fund)

(if hourly

(if (< 40 quantity)

(- (hourlyW/40+ quantity rate) fund)

(-(\* quantity rate) fund)

)

(cond ( (< 10000 quantity)

(- (\* quantity (divide 6)) fund) )

( (<= 1000 quantity)

(- (\* quantity (divide 2.5)) fund))

((< quantity 1000)

(- (\* (divide 1) quantity) fund))

)

)

)

(define (hourlyW/40+ num rate)

(+ (\* (- num 40) 1.5 rate) (\* rate 40) )

)

**Program#7 and tree diagram**

;Christopher McCoy, this program shows how to access each item from a list in scheme.

(define x'((("a" "b" "c") "d" ("e" "f")) "g" ((("h") "i"))))

(car(car(car x)))

(car(cdr(car(car x))))

(car(cdr(cdr(car(car x)))))

(car(cdr(car x)))

(car(car(cdr(cdr(car x)))))

(car(cdr(car(cdr(cdr(car x))))))

(car(cdr x))

(car(car(car(car(cdr(cdr x))))))

(car(cdr(car(car(cdr(cdr x))))))

Program#6

;Christopher McCoy, this program demonstrates how to loop in scheme by using recursion.

(define (seq initial difference num)

(if (> num 0)

(+ difference (seq initial difference (- num 1)))

initial

)

)

;(seq 2 2 5)

(define (power base x)

(if (> x 0)

(if (even? x)

(square (power base (/ x 2) ))

(\* base (square (power base (/ (- x 1) 2) )) )

)

1

)

)

(define(square x)

(\* x x)

)

; (power 2 2)

**Program#8**

#include <iostream>

void intro();

char\* userInput();

int findCharacter(char \*input,char \*key);

using namespace std;

/\*Christopher McCoy, this program is designed to read a line of code and then count

how many times a user-specified character appears in the text.\*/

int main()

{

char \*key;

char \*input;

char again,redo;

int counter;

do

{

intro();

input=userInput();

do

{

cout<<endl<<"Please enter a character you want to count how many times it appears "<<endl;

cout<<"in the line of text you entered."<<endl;

cin>>key;

counter=findCharacter(input,key);

cout<<"The character '"<<key<<"' appears "<<counter<<" times."<<endl;

cout<<"Do you want to look for another character (y/n)?"<<endl;

do

{

cin>>again;

if(again != 'y' && again !='n')

{

cout<<"Please enter (y/n)."<<endl;

}

}while(again != 'y' && again !='n');

}while(again=='y');

cout<<"Do you want to start over (y/n)?"<<endl;

do

{

cin>>redo;

if(redo != 'y' && redo !='n')

{

cout<<"Please enter (y/n)."<<endl;

}

}while(redo != 'y' && redo !='n');

cin.get();//get rid of the rest.

}while(redo=='y');

}

void intro()

{

cout<<"This program is designed to read a line of code and then count how "<<endl;

cout<<"many times a user-specified character appears in the text."<<endl;

cout << "Please enter a line of text you want to test on." << endl;

}

char\* userInput()

{

char c;

char \*input = new char[81];

c=cin.get();

char \*start=input;

while(c!='\n')

{

if(c!= ' ' && c!= '\t')

{

\*input=c;

input++;

c=cin.get();

}

else{

c=cin.get();

}

}

\*input='\0';

int length;

length=input-start;

input=start;

char \*newArray=new char[length];

start=newArray;

while(\*input!='\0')

{

\*newArray=\*input;

newArray++; input++;

}

newArray=start;

return newArray; //shift back to start

}

int findCharacter(char \*input,char \*key)

{

int counter=0;

while(\*input!='\0')

{

if(\*input==\*key)

{

++counter;

}

input++;

}

return counter;

}