**CS120 Fundamental of Programming**

**Homework 5: Repetition Structure**

**Example 1:**

In this example, the user is prompted to enter a number. The task is repeated until the user enters number 0. The entire list of numbers is then added and the sum is displayed. A sample run of the program is shown:

This program sums a series of integers.

Enter a number (0 to terminate): 8

Enter a number (0 to terminate): 23

Enter a number (0 to terminate): 71

Enter a number (0 to terminate): 5

Enter a number (0 to terminate): 0

The sum is: 107

Again, we need to focus about the tasks that need to be repeated. We can observe that input task is repeated. Moreover, a closer look indicates that every input number is added repeatedly to accumulate the final sum. The code below uses these observations to sum the numbers received input from the user.

/\* Sums a series of numbers \*/

#include <stdio.h>

int main(void)

{

int n, sum = 0;

printf("This program sums a series of integers.\n");

printf("Enter integers (0 to terminate): ");

scanf("%d", &n);

while (n != 0) {

sum += n;

scanf("%d", &n);

}

printf("The sum is: %d\n", sum);

return 0;

}

**Example 2:**

/\* Calculates the number of digits in an integer \*/

#include <stdio.h>

int main(void)

{

int digits = 0, n;

printf("Enter a nonnegative integer: ");

scanf("%d", &n);

do {

n /= 10;

digits++;

} while (n > 0);

printf("The number has %d digit(s).\n", digits);

return 0;

}

**Example 3:**

\*\*\* ACME checkbook-balancing program \*\*\*

Commands: 0=clear, 1=credit, 2=debit, 3=balance, 4=exit

Enter command: 1

Enter amount of credit: 1042.56

Enter command: 2

Enter amount of debit: 133.79

Enter command: 1

Enter amount of credit: 1754.32

Enter command: 2

Enter amount of debit: 1400

Enter command: 2

Enter amount of debit: 68

Enter command: 2

Enter amount of debit: 50

Enter command: 3

Current balance: $1145.09

Enter command: 4

/\* Balances a checkbook \*/

#include <stdio.h>

int main(void)

{

int cmd;

float balance = 0.0f, credit, debit;

printf("\*\*\* ACME checkbook-balancing program \*\*\*\n");

printf("Commands: 0=clear, 1=credit, 2=debit, ");

printf("3=balance, 4=exit\n\n");

for (;;) {

printf("Enter command: ");

scanf("%d", &cmd);

switch (cmd) {

case 0:

balance = 0.0f;

break;

case 1:

printf("Enter amount of credit: ");

scanf("%f", &credit);

balance += credit;

break;

case 2:

printf("Enter amount of debit: ");

scanf("%f", &debit);

balance -= debit;

break;

case 3:

printf("Current balance: $%.2f\n", balance);

break;

case 4:

return 0;

default:

printf("Commands: 0=clear, 1=credit, 2=debit, ");

printf("3=balance, 4=exit\n\n");

break;

}

}

}

This example utilizes for loop such that it may appear very confusing. The for loop above (for (;;)) is missing any initialization, condition, or loop variable change expression. This instance of for loop (for (;;)) is considered as an infinite loop which is true forever. Thus, the loop body has to take control of the action such that the loop body repetition terminates when it meets a condition. In the above example, the ‘case 4’ of switch structure is an example of this. When the user inputs 4, ‘case 4’ is true and ‘return 0’ statement is executed which terminates the program by returning control to the operating system.

**Exercises**

1. Write a C program to compute factorial of a number using for loop. Factorial is represented using '!', so five factorial will be written as (5!), n factorial as (n!). Also n! = n\*(n-1)\*(n-2)\*(n-3)...3.2.1 and zero factorial is defined as one i.e. 0! = 1.
2. The harmonic mean is another way of calculating the mean for a set of numbers. The harmonic mean of a set of numbers is given by the equation  
   Write a program that will input 10 number (i.e. N = 10) of positive input values and calculate the harmonic mean of the numbers.
3. Write a program that inputs a number and prints its square and cube values. The program should repeat this process, unless the input number is a negative value.
4. Write a program to find the range of a set of n numbers. The numbers are input through the keyboard via a loop. Range is the difference between the smallest and biggest numbers in the list. Hint: Inside the loop, find the minimum and the maximum number.