Software Engineering Department

Computer Organisation and Programming Course
final assignment

Pocket Calculator application

Written by:

Adham jaber 204218184

&

Ayham hussein 314813270

10 August 2020

Lecturer: Dr Yigal Hoffner

1.**introduction**

mano pocket calculator supports 4 calculations (+,-,*,/) range between (-32767-32768)
enter number operation and second number and hit carriage return .

2. extra work carried out (25%)

- Check input and output overflow or underflow 5%
- Back-space in input 5%
- Fast multiplication and fast division (binary) 10%
- Sophisticated input 10%

3. major design and implementation

First when u input the numbers and operation is stored in a string, then there is a subroutine that takes the numbers (check if they are valid) and then calculates.
Checks if over/underflow, division by 0 ,

The most challenging part was doing the binary division and I will explain it here because it is a bit hard to write it in a high level algorithm

```
division: (dividend, divisor)

    result = 0
    remainder = 0

    for each bit in divisor:
        CIL remainder
        remainder += current bit in divisor
        CIL result
        if reaminder >= divisor
            remainder -= divisor
            result += 1

    return result, remainder
```

Users guide

Just type in what you want to calculate and hit enter
>100+4
>5000/19
…

High level code

```
Int get_num (char* input, int error, char* curr_ptr);
Int plus_fun (int num1, int num2);
Int minus_fun ( int num1, int num2);
Int multi_fun (int num1, int num2);
Int div_fun (int num1, int num2);

Int main ()
{

        While(*input != 13)                     // input!= carriage-return
        {
                Scanf("%c" , &input)
        }
        Num1=get_num(input)
        Op=get_num(input)
        Num2=get_num(input)
        If(num1> 32767 || num2 > 32767)
        {
                Printf("number too big")
        }
        If (op == '+')
                Ress=plus_fun( num1, num2)
        If (op=='-')
                Ress= plus_fun(num1, num2)
        If(op== '*')
                Ress =multi_fun (num1, num2)
        If( op=='/')
                Ress=div_fun ( num1 ,num2)

        If(res>32767)
        {
                Printf("over/under flow")
        }
        else
                Printf("%d", ress)

}
```

```
Int get_num (char* input, int* error, char* ptr ,int* minus_flag)
{
        Int i=o;

        Int cr=30H

        If(input[i]==’-‘)
        {
                Minus_flag=1
                i++
        }
        While(ptr!=’+’ && ptr != ‘-‘ && ptr!= ‘*’ && ptr!= ‘/’)
        {
                Digit=ptr[i]
                Num=num*10+digit

        }
        Return num

}
Int plus_fun (int num1, int num2)
{
        Return num1+num2
}
Int minus_fun ( int num1, int num2)
{
        Return (num1-num2)
}
Int multi_fun (int num1, int num2)
{
        Return (num1*num2)              // binary short multiplication
Int div_fun (int num1, int num2)
{
        If(num2== 0)
        {
                Printf(“math error”)
                Return  null
        }
        Return (num1/num2)              //      long binary division
```