FLOAT

CSC 236

FLOAT assignment

- Read the document
- Solve for pi
  - Using Salamin-Brent algorithm
- Given
  - The formula
  - The pseudocode
  - The assembler code
  - Well, most of the assembler code

Salamin-Brent

$$\pi = \frac{4a_n^2}{1 - \sum_{k=0}^{n} 2^k (a_n - b_n)^2}$$

$$a_0 = 1$$

$$b_0 = \frac{1}{\sqrt{2}}$$

$$a_{n+1} = \frac{1}{2}(a_n + b_n)$$

$$b_{n+1} = \sqrt{a_n b_n}$$

C code

```
float a, b, c, d, s, t, pi, old;    // variables

a   = (float)1.0;                   // a0 = 1
b   = (float)1.0 ;                  // b0 = 1
b   = b / froot ((float)2.0);       // b0 /= sqrt(2)
s   = (float)1.0;                   // s is the sum
t   = (float)1.0;                   // 2**k, k = 0
old = (float)0.0;                   // prev pi
```

```
while (1)                           // loop forever
{
  // subtract next value of sum from s
  s = s - t * (a - b) * (a - b);
  pi = 4 * a * a / s;     // new pi
  c = (a + b) / 2.0;      // an+1 = (an + bn)/2
  d = froot (a * b);      // bn+1 = sqrt(an * bn)
  a = c;                  // set an+1
  b = d;                  // set bn+1
  t = 2 * t;              // next 2**k
  output(pi);             // print current pi
  if (pi == old) break;   // stop if pi no change
  old = pi;               // set old for next iter
}
```

C code

```
float a, b, c, d, s, t, pi, old;    // variables

a   = (float)1.0;                   // a0 = 1
b   = (float)1.0 ;                  // b0 = 1
b   = b / froot ((float)2.0);       // b0 /= sqrt(2)
s   = (float)1.0;                   // s is the sum
t   = (float)1.0;                   // 2**k, k = 0
old = (float)0.0;                   // prev pi
```

Assignment code 3 lines

Approximately 26 instructions

push, pop, arithmetic

Given:

- **Square root**
- **Compare**
- **Output**

```
while (1)                           // loop forever
{
  // subtract next value of sum from s
  s = s - t * (a - b) * (a - b);
  pi = 4 * a * a / s;     // new pi
  c = (a + b) / 2.0;      // an+1 = (an + bn)/2
  d = froot (a * b);      // bn+1 = sqrt(an * bn)
  a = c;                  // set an+1
  b = d;                  // set bn+1
  t = 2 * t;              // next 2**k
  output(pi);             // print current pi
  if (pi == old) break;   // stop if pi no change
  old = pi;               // set old for next iter
}
```

Steps

1. Design solution
2. Code solution
    - Retrieve UNPACK.EXE from float locker
    - Place in P23X/FLOAT directory
    - Run UNPACK.EXE in DOSbox
    - Rename *float.m* to *float.asm*
    - Put your code in *float.asm*
    - Link *float.obj* with *sqroot.obj* and *output.obj*

Steps

3. Test and debug solution
    - Test type: *float*
    - Output should be:

```
4.37534
3.18879
3.14168
3.14159
3.14159
```

Steps

4. Grading
    - To grade type: *gradfl*
    - Grade: 100% on correct answers
5. Submit assignment
    - Upload *float.ans*