Robot Programming

Zhidong Wang

Oct. 26, 2021

Project 2
Build a Robo-Kick Simulator

Skills on implementation algorithms simulating with dynamics of a robot and a ball

- 2D Runtime Graphics
- Simulation of a Ball with Dynamics
- Implement Data in Structure Style
- Simulation of a Robot Leg with Dynamics and Control
- Simulation of Interaction among robot leg, ball and environment

Project 2
Build a Robo-Kick Simulator

Skills on implementation algorithms simulating with dynamics of a robot and a ball

- 2D 実時間グラフィックス環境
- ダイナミックスを考慮したボールのダイナミクス
- 構造化のデータスタイルでの実装
- ダイナミックスと制御を考慮したロボットの脚部のシミュレーション
- ロボットの脚部とボール、そして環境との相互作用のシミュレーション

A Nice Simulator with Dynamics

Step by Step

toward Robo-Kick Simulator

- Build the 2D graphics environment
- Show a Ball moving on the ground
- Simulate the Ball moving with friction force and gravity
- Simulate the Ball free-flying and bouncing
- Simulate the Ball kicked by a Foot
- Simulate the motion of the Leg and the Ball when the Leg kicks the Ball

http://www.robotics.it-chiba.ac.jp/wang/lect/

On Steps

toward Robo-Kick Simulator

Essential Technologies

Data + Modeling + Graphics(CG)

Step by Step : Graphics

toward Robo-Kick Simulator

- Build the 2D graphics environment
  by using OpenGL and GLUT Library

http://www.opengl.org

Instruction and Source Files Download

http://www.robotics.it-chiba.ac.jp/wang/lect/

Step by Step : Graphics

toward Robo-Kick Simulator

- Build the 2D graphics environment
  by using OpenGL and GLUT Library

http://www.opengl.org

Instruction and Source Files Download

http://www.robotics.it-chiba.ac.jp/wang/lect/

For Windows

glut32.dll

32bitPC ☛ Windows内System32にコピー

64bitPC ☛ Windows内SysWoW64にコピー

Computer Graphics (1)

sample1.c をダウンロード、内容を理解し、改造する

- 四角形状を描画する部分を改造して、多角形を作成・描画してみる
- 等辺３６角形を作成し、円を近似的に描画する

  （ for 文を利用する）
- チャレンジトピック：円の描画部分を改造し、パックマン（Pac-Man)を作成してみる

Computer Graphics (2)

sample3.cをダウンロード実行し、内容とある程度理解し、改造する

（マウスの左ボタンと右ボタンをクリックしてみる）

- 四角形の描画の部分をsample1.cの描画部分に置き換えて、多角形か円形に描画できるようにする
- 正方形か円形を横に移動できるようにする

On Steps

toward Robo-Kick Simulator

Essential Technologies

Data + Modeling + Graphics(CG)

Data

- **Data Main-Body** **データ本体**
- Properties, Status 特性、状態
- Associated Links 関連情報リンク

Type of Data in C

		Format in scanf and printf
• 整数	int	%d
• 浮動小数	float	%f
• 倍精度浮動小数	double	%lf
• 文字	char	%c
• 文字列	char[ ]	%s

- 配列
  - int vec[] = {0, 0, 1};
  - char sp[] = “Thank you.”;
  - double a[4][3] ;

Type of Data in C

		Format in scanf and printf
• 文字	char	%c
• 文字列	char[ ]	%s

- 配列
  - int vec[] = {0, 0, 1};
  - char sp[] = “Thank you.”;
  - double a[4][3]

  - int vec[3];
  - vec[0] = 0; vec[1]=0; vec[2]=1;

Data Type: Struct 構造体

Represent data as a structure

```
struct  complex {
        double      re;
        double      im;
};
```

構造体の定義

```
    double
struct  complex  a, b;

a.re  = 1.0;
a.im =  2.3;

b.re  =  a.re + 2.0;
b.im  =  a.im – 3.0;
```

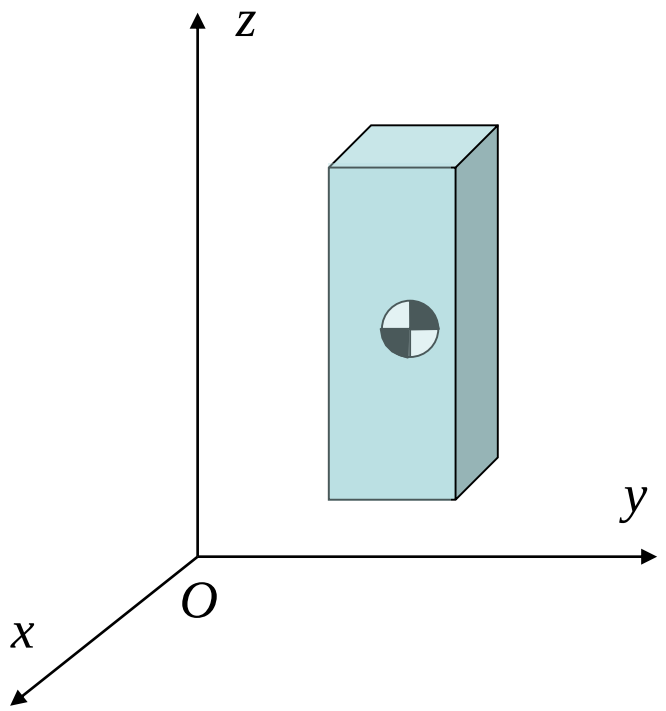
The better thing to use data type: Struct

```
struct  rigidBody {
        double        x,     y,     theta;
        double              dx,    dy,   dtheta;
        double      ddx, ddy, ddtheta;
};

struct  rigidBody   foot;

foot.x  = 1.0;       foot.y  =  0.0;
foot.theta = 90.0;

foot.dx  = 0.5;      foot.dy  =  1.0;
foot.dtheta = -5.0;
.......
```



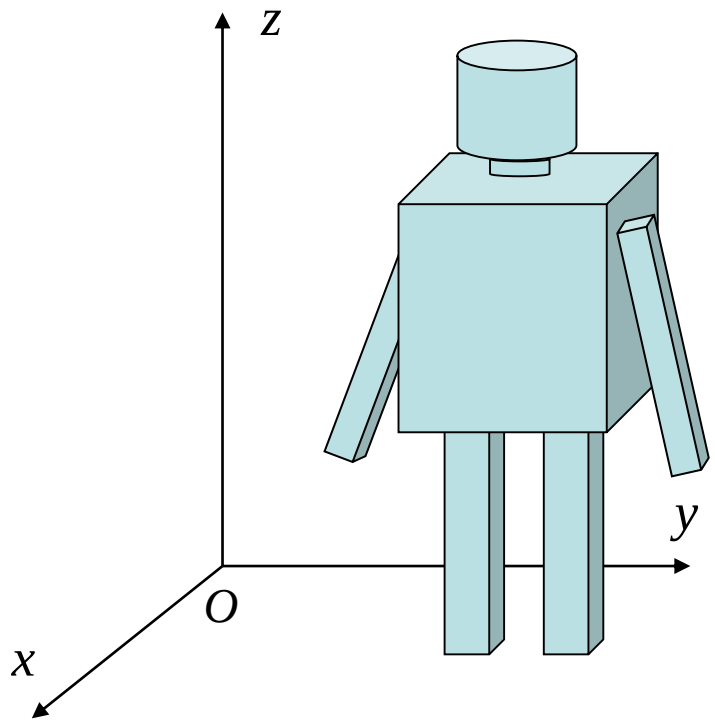
The better thing to use data type: Struct

```
struct  rigidBody {
          double        x,     y,     theta;
          double       dx,    dy,   dtheta;
          double      ddx, ddy, ddtheta;
};

struct   robot {
          int     id;
          struct  rigidBody     head,  body;
          struct  rigidBody     armL,  armR;
          struct  rigidBody     legL,  legR;
};

strcut   robot    rbt1;

rbt1.id = 1;
rbt1.legL.x  = 1.0;       rbt1.armR.y  =  3.0;
…….
```



Data Type: Struct

```
struct  person {
        int      id;
        char   name[40];
        long   phone;
};

struct  person  student;

printf("%d  %s  %ld¥n",  student.id,
                         student.name,
                         student.phone);
```

Sub Project

Build a program which has 5 persons' data with struct type. The program will print out all 5 persons' information.

Data Type: Struct

```
struct  person {
        int      id;
        char   name[40];
        long   phone;
};

struct  person  student;

scanf("%d  %s  %ld",  &student.id,
                      student.name,
                      &student.phone);
```

Sub Project

Build a program which has 5 persons' data with struct type. The program will print out all 5 persons' information.

What you should learn from Project 1 (Build a Multi-Function Calculator)

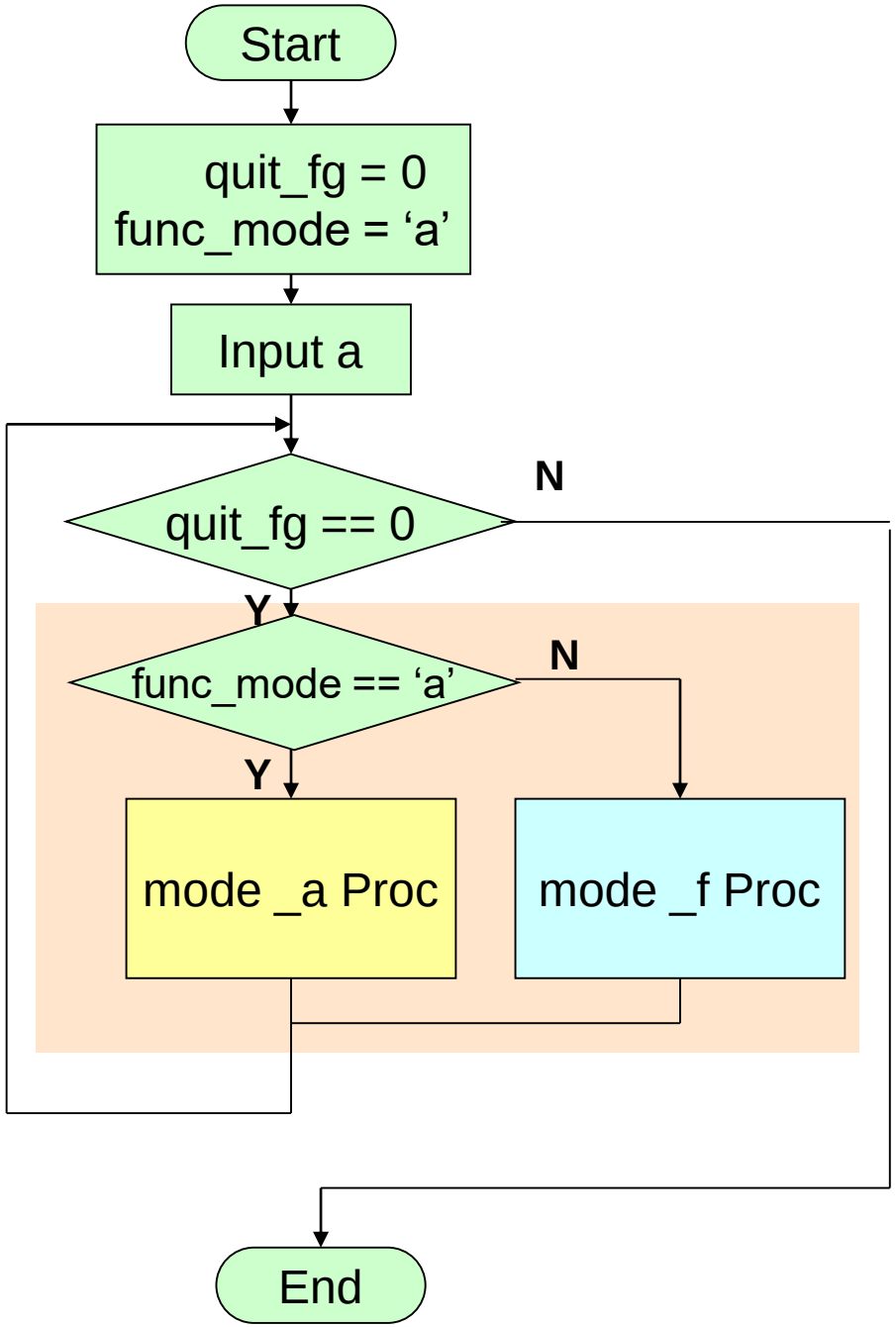
Skills on implementation algorithms

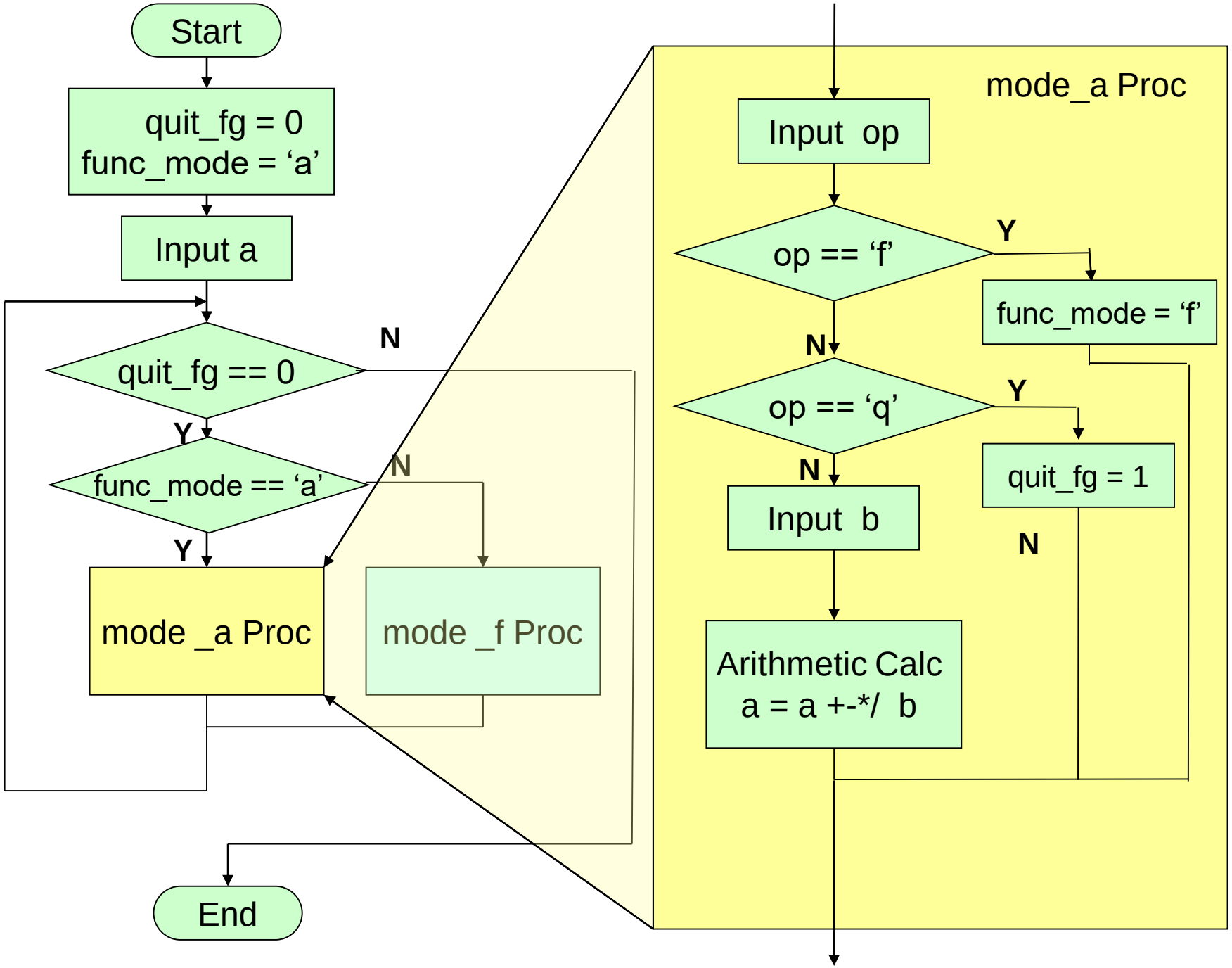
- Basic Functions
  - scanf, printf
  - for, while
  - if, switch
  - math.h, sin, cos, log
  - Data Type (int, float, double, char)

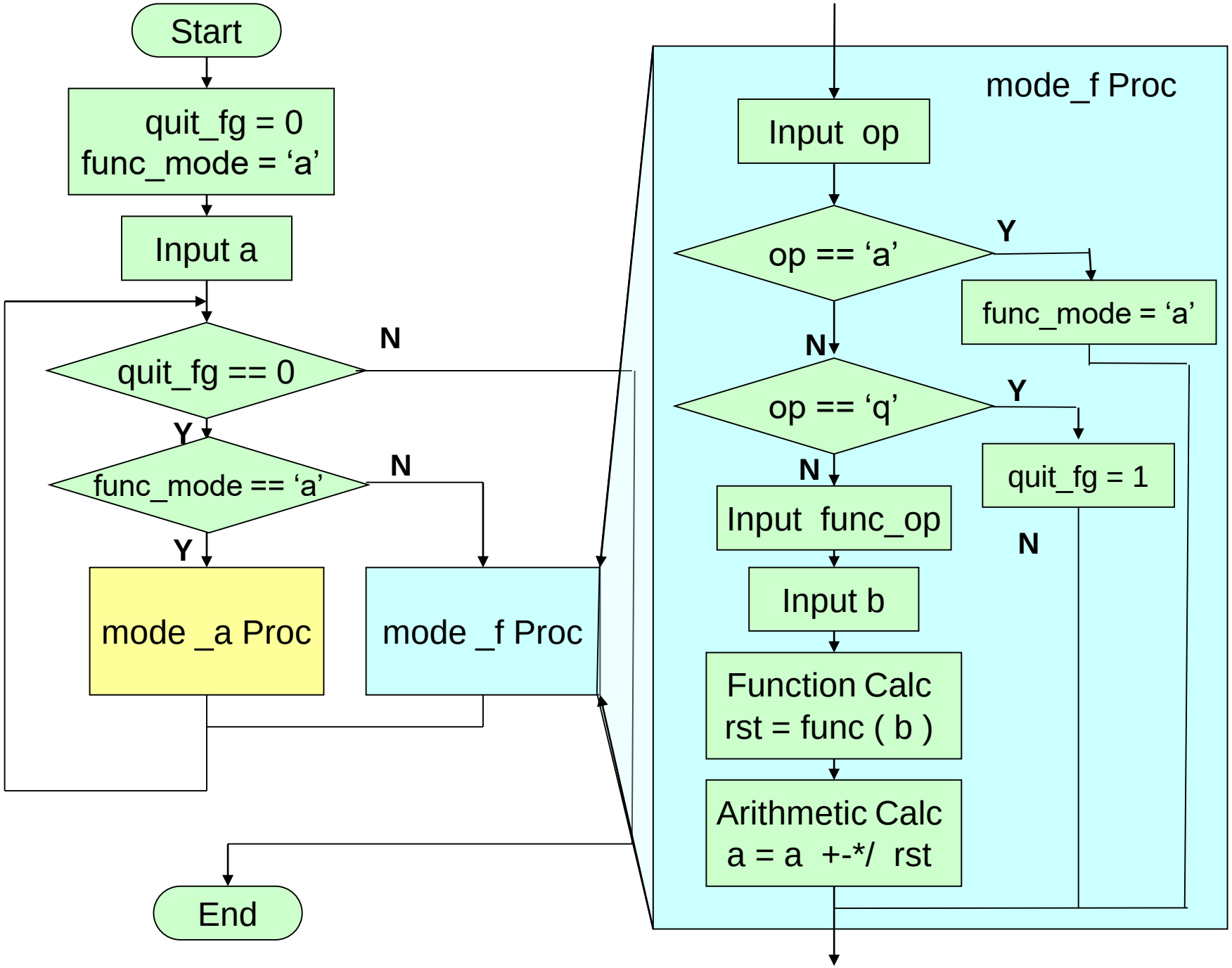
What you should learn from Project 1 (Build a Multi-Function Calculator)

Skills on implementation algorithms

- Flag technique and implementation of multi-mode system
  - Using a Boolean type or Integer type data as a flag
  - By changing the flag in the program, you can control the flow of the program


Start
quit_fg = 0
func_mode = ‘a’
Input a
quit_fg == 0
N
Y
func_mode == ‘a’
N
Y
mode _a Proc
mode _f Proc
End


Start
quit_fg = 0
func_mode = ‘a’
Input a
quit_fg == 0
N
Y
func_mode == ‘a’
N
Y
mode _a Proc
mode _f Proc
End
mode_a Proc
Input op
op == ‘f’
Y
func_mode = ‘f’
N
op == ‘q’
Y
quit_fg = 1
N
N
Input b
Arithmetic Calc
a = a +-*/ b


Start
quit_fg = 0
func_mode = ‘a’
Input a
quit_fg == 0
N
Y
func_mode == ‘a’
N
Y
mode _a Proc
mode _f Proc
End
mode_f Proc
Input op
op == ‘a’
Y
func_mode = ‘a’
N
op == ‘q’
Y
quit_fg = 1
N
N
Input func_op
Input b
Function Calc
rst = func (b)
Arithmetic Calc
a = a +-*/ rst

What you should learn from Project 1

(Build a Multi-Function Calculator)

Skills on implementation algorithms

- Structured Programming
  - Make your program easy to understand
  - Do not use “goto”

    Let the flow of your program be simple
  - Better to Use “while” than “do – while”

    Provide the condition in the beginning