

課題 1

問題文

課題 `int add2(int x, int y) { return (x+y); }` と関数が定義され、`main` 関数内で `int num1=2; int num2=3;` と宣言された時、式 `add2(num1,num2)` の評価値は 5 で あった。その評価値の正しさは、以下のような部 分式の評価の列を示すことで示される。`add2(num1,num2) → add2(2,3) → 5`

同じ定義と宣言と、宣言 `int num3;` のもとで、式 `num3=add2((3+num1)*num2,num2*num2)` について、その評価値とその部分式の評価の列を 書け。

Source code

```
#include <stdio.h>

int add2(int x, int y);
int main(void)
{
    int num1 = 2;
    int num2 = 3;
    int num3;

    add2(num1, num2);
    num3 = add2((3 + num1) * num2, num2 * num2);
    printf("num3:%d", num3);

    return 0;
}

int add2(int x, int y)
{
    return (x + y);
}
```

```
Last login: Fri Nov 16 21:03:51 on ttys002
dot1x7310:~ admin$ /Users/admin/Documents/GitHub
;
num3:24logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

課題 2

問題文

課題 次の関数を定義したプログラムは、

```
#include int mul2(int num){ return (2*num); } int main(void){ int result; result=mul2(3);  
printf("%d\n",result); }
```

次のプログラムのように関数を用いない 形に書きなおせる。

```
#include int main(void){ int result; int num=3; result=(2*num); printf("%d\n",result); }
```

課題(つづき) 次の関数を定義したプログラムを、関数を用いない形に書き直せ。

```
#include int remainder(int num1, int num2){ return (num1%num2); } int main(void){ int  
result; result=remainder(123,10); printf("%d\n",result); }
```

Source code

```
#include <stdio.h>  
  
int main(void)  
{  
    int num1 = 123;  
    int num2 = 10;  
    int result;  
  
    result = num1 % num2;  
    printf("%d", result);  
  
    return 0;  
}
```

```
Last login: Fri Nov 16 21:20:26 on ttys002  
dot1x7310:~ admin$ /Users/admin/Documents/GitHub  
;  
3logout  
Saving session...  
...copying shared history...  
...saving history...truncating history files...  
...completed.  
█
```

課題 3

問題文

課題 次のプログラムと `int main(void){ int result; int num1=123; int num2=10; result=(num1/num2); printf("%d\n",result); }` 次のプログラムが同じような動作をするように関数 `quotient` を定義せよ。 `int main(void){ int result; result=quotient(123,10); printf("%d\n",result); }`

Source code

```
#include <stdio.h>

int quotient(int x, int y);

int main(void)
{
    int result;

    result = quotient(123, 10);

    printf("%d", result);

    return 0;
}

int quotient(int x, int y)
{
    return (x / y);
}
```

```
Last login: Fri Nov 16 21:22:25 on ttys
dot1x7310:~ admin$ /Users/admin/Documen
;
12logout
Saving session...
...copying shared history...
...saving history...truncating history
...completed.

[プロセスが完了しました]
```

課題 4-1

問題文

課題 ・ 次のプログラムの下線部を関数呼 び出しにせよ。関数も定義せよ。

```
#include <stdio.h>
int main(void){ int a,b,c,x,y;; printf("整数 A:"); scanf("%d",&a); printf("整数 B:"); scanf("%d",&b); x=a; y=b; if ( x > y) c=x; else c=y; printf("大きいほうは%d\n",c); }
```

Source code

```
#include <stdio.h>
int comparison(int x, int y);
int main(void)
{
    int a, b, c;
    printf("整数 A:");
    scanf("%d", &a);
    printf("整数 B:");
    scanf("%d", &b);
    c = comparison(a, b);
    printf("大きいほうは%d\n", c);

    return 0;
}
int comparison(int x, int y)
{
    if (x > y)
        return (x);
    else
        return (y);
}
```

```
Last login: Fri Nov 16 21:25:23 on ttys002
dot1x7310:~ admin$ /Users/admin/Documents/GitHub
it;
整数 A:50
整数 B:20
大きいほうは 50
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

課題 4-2

問題文

課題

・ 次のプログラムの下線部を関数呼び出しにせよ。関数も定義せよ。 #include <stdio.h>
main(void){ int a,b,c; int i=0; printf("整数:"); scanf("%d",&a); b=a;
while((b%2==0)&&(b!=0)){ b/=2; i++; } c=i; printf("整数%d は 2 で、%d 回割れる
¥n",a,c); }

Source code

```
#include <stdio.h>

int div(int x);

int main(void)
{
    int a, b, c;
    int i = 0;
    printf("整数:");
    scanf("%d", &a);
    c = div(a);
    printf("整数%d は 2 で、%d 回割れる¥n", a, c);

    return 0;
}

int div(int x)
{
    int itr;

    while ((x % 2 == 0) && (x != 0))
    {
        x /= 2;
        itr++;
    }

    return (itr);
}
```

```
Last login: Fri Nov 16 21:30:39 on ttys002
dot1x7310:~ admin$ /Users/admin/Documents/GitHub/
it;
整数:40
整数 40 は 2 で、 3 回 割 れ る
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

課題 5

問題文

課題 正の実数を入力としその実数の小数部分を返却する関数 dpart を定義せよ。

```
int main(void){ double dnum; printf("実数を入力してください:"); scanf("%lf",&dnum);  
printf("%f の小数部分は%f\n", dnum, dpart(dnum)); return 0; }
```

動作例 >実数を入力
してください:3.14 > 3.14 の小数部分は 0.14 >

Source code

```
#include <stdio.h>  
  
double dpart(double x);  
  
int main(void)  
{  
    double dnum;  
    printf("実数を入力してください:");  
    scanf("%lf", &dnum);  
    printf("%lf の小数部分は %lf\n", dnum, dpart(dnum));  
  
    return 0;  
}  
  
double dpart(double x)  
{  
    return x - (int)x;  
}
```

```
Last login: Fri Nov 16 21:30:58  
dot1x7310:~ admin$ /Users/admin  
;  
実数を入力してください:4.77777  
4.777770の小数部分は 0.777770  
logout  
Saving session...  
...copying shared history...  
...saving history...truncating  
...completed.  
  
[プロセスが完了しました]
```

レポート課題（自作問題） 1-1

問題文

time.h ヘッダを使用して inline 関数の実行速度の変化を測定し表示する。

For 文で 1000 万回変数を足し算させる動作を inline 関数と通常関数で実行

Source code

```
#include <stdio.h>
#include <time.h>

inline int fun_inline(int x, int y);
int fun(int x, int y);
int sum(int x, int y);
int main(void){
    time_t start, end;
    int a = 10;
    int b = 90;
    start = clock();
    for (int i = 0; i < 1000000000; i++){
        fun(a, b);
    }
    end = clock();
    printf("通常の関数の実行時間:%.8f\n", (double)(end - start) / CLOCKS_PER_SEC);
    start = clock();
    for (int i = 0; i < 1000000000; i++){
        fun_inline(a, b);
    }
    end = clock();
    printf("inline 関数の実行時間:%.8f\n", (double)(end - start) / CLOCKS_PER_SEC);
}

inline int fun_inline(int x, int y){
    return x + y;
}

int fun(int x, int y){
    return x + y;
}
```

```
Last login: Sat Nov 17 01:08:24
C02SM7FSGTFJ:~ admin$ /Users/adm
通常の関数の実行時間:0.00000200
inline関数の実行時間:0.00000100
logout
Saving session...
...copying shared history...
```

レポート課題（自作問題） 1-2

問題文

1-1 の問題で変数を `register int` にした場合の実行時間の変化と 1-1 の実行結果を、十回実行した時の平均を値として出力する

※コンパイル時にコンパイラオプションにて `clang` コード生成オプション、最適化レベルを 3 に設定してコンパイルする必要あり。(1-1 でも同様)

Source code

```
#include <stdio.h>
#include <time.h>

//関数プロトタイプ宣言部
inline int fun_inline(int x, int y);
int fun(int x, int y);
double average(double x[]);

int main(void)
{
    //変数宣言部
    time_t start, end;
    int i = 0;
    int j = 0;
    int a1 = 10;
    int b1 = 90;
    double avg = 0;
    register int a2 = 10;
    register int b2 = 90;
    double num_int[10];
    double num_registerint[10];

    //通常関数かつ非 register 変数での処理部
    for (i = 0; i < 10; i++)
    {
        start = clock();
        for (j = 0; j < 1000000000; j++)
        {
```



```

        fun(a1, b1);
    }

    end = clock();
    num_int[i] = ((double)(end - start) / CLOCKS_PER_SEC);
}

avg = average(num_int);
printf("通常関数かつ非 register 変数での処理平均値:%.20f\t¥n", avg);
avg = 0;

//inline 関数かつ非 register 変数での処理部
for (i = 0; i < 10; i++)
{
    start = clock();
    for (j = 0; j < 1000000000; j++)
    {
        fun_inline(a1, b1);
    }
    end = clock();
    num_int[i] = ((double)(end - start) / CLOCKS_PER_SEC);
}

avg = average(num_int);
printf("inline 関数かつ非 register 変数での処理平均値:%.20f\t¥n", avg);

//通常関数かつ register 変数での処理部
for (i = 0; i < 10; i++)
{
    start = clock();
    for (j = 0; j < 1000000000; j++)
    {
        fun_inline(a2, b2);
    }
    end = clock();
    num_int[i] = ((double)(end - start) / CLOCKS_PER_SEC);
}

avg = average(num_int);
printf("通常関数かつ register 変数での処理平均値:%.20f\t¥n", avg);

```

```

//inline 関数かつ register 変数での処理部
for (i = 0; i < 10; i++)
{
    start = clock();
    for (j = 0; j < 1000000000; j++)
    {
        fun_inline(a2, b2);
    }
    end = clock();
    num_int[i] = ((double)(end - start) / CLOCKS_PER_SEC);
}

avg = average(num_int);
printf("inline 関数かつ register 変数での処理平均値:%.20f\t¥n", avg);
}

inline int fun_inline(int x, int y)
{
    return x + y;
}

int fun(int x, int y)
{
    return x + y;
}

double average(double x[])
{
    double sum = 0;
    double avg = 0;
    for (int i = 0; i < 10; i++)
    {
        sum += x[i];
    }
    avg = sum / 10;
    return avg;
}

```

```
Last login: Tue Nov 20 16:06:38 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub/programming-report
通常関数かつ非register変数での処理平均値:0.00000170000000000000
inline関数かつ非register変数での処理平均値:0.00000130000000000000
通常関数かつregister変数での処理平均値:0.00000130000000000000
inline関数かつregister変数での処理平均値:0.00000120000000000000
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

課題レポート 2

問題文

以下のような関数頭部をもち、引数 x と y をかけた値を返却 する関数を書け。 `int multi(int x, int y)` その関数を呼び出す `main` 関数と、実行結果も書け。

Source code

```
#include <stdio.h>

int multi(int x, int y);

int main(void)
{
    int num1 = 10;
    int num2 = 10;

    printf("num1*num2=%d", multi(num1, num2));

    return 0;
}

int multi(int x, int y)
{
    return x * y;
}
```

```
Last login: Tue Nov 20 16:07:13 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub
num1*num2=100logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

課題レポート 3

問題文

以下のような関数頭部をもち、引数 x の10進での1桁目を 返す関数を書け。 `int lowest(int x)` その関数を呼び出す `main` 関数と、実行結果も書け。

Source code

```
#include <stdio.h>

int lowest(int x);

int main(void)
{
    int num = 123456789;
    int low;

    low = lowest(num);
    printf("123456789 の一桁目は%d", low);

    return 0;
}

int lowest(int x)
{
    return x % 10;
}
```

```
Last login: Tue Nov 20 16:19:34 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub
123456789の一桁目は9logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

課題レポート 4

問題文

以下のような関数頭部をもち、引数 x, y, z の平均値を返す関 数を書け。 `double avrg(double x, double y, double z)` その関数を呼び出す `main` 関数と、実行結果も書け。

Source code

```
#include <stdio.h>

double avrg(double x, double y, double z);

int main(void)
{
    double x = 123;
    double y = 456;
    double z = 789;
    double avg;

    avg = avrg(x, y, z);
    printf("平均:%lf", avg);

    return 0;
}

double avrg(double x, double y, double z)
{
    return (x + y + z) / 3;
}
```

```
Last login: Tue Nov 20 16:20:45 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub
平均:456.000000logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

課題レポート 5

問題文

以下のような関数頭部をもち、キーボードから入力された整数を2倍した値を返す関数を書け。int get2(void) その関数を呼び出す main 関数と、実行結果も書け。

Source code

```
#include <stdio.h>
int get2(void);
int main(void)
{
    int num;
    num = get2();
    printf("二倍:%d¥n", num);

    return 0;
}
int get2(void)
{
    int get;
    scanf("%d", &get);

    return get * 2;
}
```

```
Last login: Tue Nov 20 16:22:52 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub
50
二倍 :100
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```

課題レポート 6

問題文

定義した関数がどのようにメモリ上に配置されるかを調べ、その様子を図を用いて説明せよ。

Source code

```
#include <stdio.h>

int fct(int x)
{
    return x + 1;
}

int fct2(int x)
{
    return x + 2;
}

int main(void)
{
    int a[3] = {1, 2, 3};
    printf("%d,%d,%d\n", a[0], a[1], a[2]);
    printf("%p,%p,%p\n", &a[0], &a[1], &a[2]);
    printf("%d\n", fct(2));
    printf("%p\n", fct);
    printf("%d\n", fct2(2));
    printf("%p\n", fct2);
    return 0;
}
```

```
Last login: Tue Nov 20 16:24:14 on ttys003
dot1x0926:~ admin$ /Users/admin/Documents/GitHub
1,2,3
0x7ffee85fab1c,0x7ffee85fab20,0x7ffee85fab24
3
0x107605e10
4
0x107605e30
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[プロセスが完了しました]
```


説明

変数は4 飛ばし、関数は2 0 飛ばしにメモリに格納されていることがわかった。