Xilinx Zynq FPGA, TI DSP, MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)
gcccompil3r@gmail.com
학생 - 이유성
dbtjd1102@naver.com

x^2함수의 미분

```c
#include<stdio.h>
#include<math.h>

#define dx 0.001

double f_x(double x);

double diff(double (*p)(double),double x);

int main(void)
{
        double x,ret ;
        double (*p)(double);
        p = f_x;

        double val;
        printf("input\n");
        scanf("%lf",&val);

        ret = diff(p,val);
        printf("ret = %lf\n ",ret);

        return 0;

}

double f_x(double x)  //y = x^2
{
        double y;
        y = pow(x,2);

        return y;
}

double diff(double (*p)(double),double x)
{
        double ret;
        ret = p(x+dx)-p(x); //differential
        ret = ret/dx;

        return ret;
}
```

x^2함수의 적분

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

#define dx 0.001


double f_x(double x);

double integral(double (*p)(double),double x1,double x2);


int main(void)
{
        double x1,x2,ret ;
        double (*p)(double);
        p = f_x;

        printf("input x1,x2\n");
        scanf("%lf %lf" ,&x1 , &x2);

        ret = integral(p,x1,x2);
        printf("ret = %lf\n",ret);

        return 0;
}

double f_x(double x)
{
        double y;
        y = pow(x,2);

        return y;
}

double integral(double (*p)(double),double x1,double x2)
{
        double ret = 0;
        int n;

        if(x1 < x2){
                for(n = 1 ; x1+n*dx <= x2 ; n++){
                        ret += dx * p(x1+n*dx) ;
                }
        }


        else {
                for(n = 1 ; x2+n*dx <= x1 ; n++){
                        ret += dx * p(x2+n*dx) ;
                }
        }
        return ret;
}
```

y = 3*e^-x^2 오차율 계산 (이상하게 짰음..미완성;;;;;;;;)

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

#define e 2.71828
#define d_x 0.001
#define start -5
#define end 5

double f_x(double x);
void diff_func(double (*p)(double),double *y);
double y_prime(double x, double (*p)(double));
void set_accuracy(double (*p)(double),double *y);


int main(void)
{
        double x = 0;
        double y[10000] = {0};
        double acc[10000] = {0};

        int i;

        double (*p)(double) ;
        p = f_x;

        for(i = 0 ; i <10000 ; i++){
                printf("%lf\n", y[i]/acc[i]);
        }

        return 0;
}

double y_prime(double x,double (*p)(double))
{
        return -2*x*p(x) ;

}

double f_x(double x)
{
        return 3*pow(e,-pow(x,2));
}

void diff_func(double (*p)(double),double *y)
{
        int i;
        for(i = start ; i = end ; i += d_x)
                y[(i+5)*1000] = (f_x( i + d_x) - f_x(i)) / d_x ;
}

void set_accuracy(double (*p)(double),double *y)
{
        int i;
        double x;
        for(x = start ; x <=end;  x +=d_x){
                i = (x+5)*1000;
                y[i] = y_prime(x,p);

                if(y[i] < 0 )
                        y[i] *=(-1);

        }
}
```