Homework #1

2015004302 곽상원

method1:

<Code>- machar()의 일부코드+main함수코드.

1) float(single precision) -machar_float.c

```
#define CONV(i) ((float)(i))

void machar(int *ibeta, int *it, int *irnd, int *ngrd, int *machep, int *negep,
int *iexp, int *minexp, int *maxexp, float *eps, float *epsneg,
float *xmin, float *xmax)
{
    int i,itemp,iz,j,k,mx,nxres;
    float a,b,beta,betah,betain,one,t,temp,temp1,tempa,two,y,z,zero;
```

```
int main() {
        int ibeta, it, irnd, ngrd, machep, negep, iexp, minexp, maxexp;
        float eps, epsneg, xmin, xmax;
        machar(&ibeta, &it, &irnd, &ngrd, &machep, &negep,
                &iexp, &minexp, &maxexp, &eps, &epsneg,
                &xmin,&xmax);
        printf("%E\n", eps);//machine eps을 출력.
        return 0;
}
```

2)double(double precision) -machar_double.c

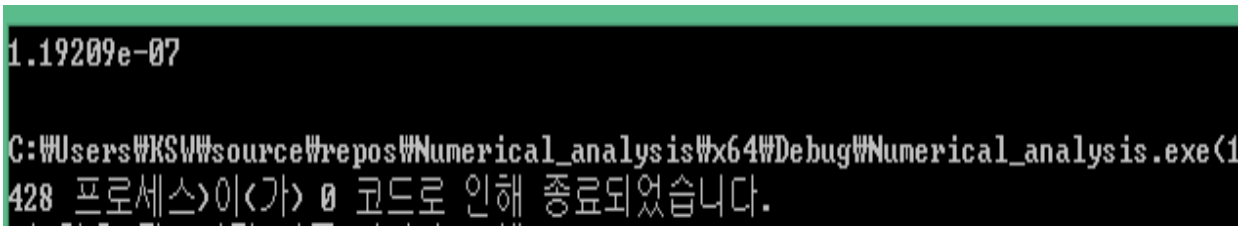
```
#define CONV(i) ((double)(i))

void machar(int *ibeta, int *it, int *irnd, int *ngrd, int *machep, int *negep,
int *iexp, int *minexp, int *maxexp, double *eps, double *epsneg,
double *xmin, double *xmax)
{
    int i,itemp,iz,j,k,mx,nxres;
    double a,b,beta,betah,betain,one,t,temp,temp1,tempa,two,y,z,zero;
```

```
int main() {
        int ibeta, it, irnd, ngrd, machep, negep, iexp, minexp, maxexp;
        double eps, epsneg, xmin, xmax;
        machar(&ibeta, &it, &irnd, &ngrd, &machep, &negep,
                &iexp, &minexp, &maxexp, &eps, &epsneg,
                &xmin,&xmax);
        printf("%E\n", eps);//machine eps을 출력.
        return 0;
}
```

<실행결과>

1)float



2)double

method2:

<Code> -Machine_accuracy.c

```
#include <stdio.h>
int get_eps_float();
int get_eps_double();

int main() {
	printf("Machine_accuracy of 'float': ");
	printf("%d₩n", get_eps_float());// float의 경우

	printf("-----------------------------------------₩n");

	printf("Machine_accuracy of 'double': ");
	printf("%d₩n", get_eps_double());// double의 경우

	return 0;
}

int get_eps_float() {
	float data = 1.0;
	float temp = 1.0, operand = 0.5;//
	int n = 0;
	while (1) {
		temp *= operand, n++;
		if (data + temp == data) break;
	}
	return n - 1;
}

int get_eps_double() {
	double data = 1.0;
	double temp = 1.0, operand = 0.5;//
	int n = 0;
	while (1) {
		temp *= operand, n++;
		if (data + temp == data) break;
	}
	return n - 1;
}
```

*get_eps_float(), get_eps_double()는 1.0(data)+ 2^-n(temp, machine_eps) > 1.0(data)을 만족하는 최소 n을 찾아서 반환하는 함수이다.

<실행결과>

```
Machine_accuracy of 'float': 23
-----------------------------------------
Machine_accuracy of 'double': 52
```

- float의 경우 n=23이므로 machine_eps = 2^-23
- double의 경우 n=52이므로 machine_eps = 2^-52