
TP n° 7
Assembleur + C

Exercice 1 - On désire implanter une structure de vecteur de float, composée comme suit :

```
typedef struct {  
    int size; // taille maximale  
    float *data; // données allouées  
} Vector;
```

On vous demande d'implanter en assembleur les méthodes suivantes :

```
void vector_construct(Vector *v, int sz) {  
    v->size=sz;  
    v->data=(float *)calloc(sz,sizeof(float));  
}  
  
void vector_delete(Vector *v) {  
    free(v->data);  
}  
  
void vector_set(Vector *v, int ndx, float value) {  
    v->data[ndx]=value;  
}  
  
float vector_get(Vector *v, int ndx) {  
    return v->data[ndx];  
}  
  
void vector_add(Vector *x, Vector *y, Vector *z) {  
    for (int i=0;i<x->size;++i) z->data[i] = x->data[i]+y->data[i];  
}  
  
float vector_sum(Vector *v) {  
    float s=0.0;  
    for (int i=0;i<v->size;++i) s+=v->data[i];  
    return s;  
}  
  
void vector_print(Vector *v) {  
    for (int i=0;i<v->size;++i) printf("[%d]=%f\n",i,x->data[i]);  
}
```

Puis, d'utiliser les méthodes implantées dans le programme C suivant :

```
#include "vector.h"  
#define SIZE 20
```

```
int main() {
    Vector x, y, z;

    Vector_construct(&x,SIZE);
    Vector_construct(&y,SIZE);
    Vector_construct(&z,SIZE);

    for (int i=0;i<SIZE;++i) {
        Vector_set(&x,i,2.0*(i+1));
        Vector_set(&y,i,-3.0*(i+1));
    }
    Vector_add(&x, &y, &z);
    printf("sum = %f", Vector_sum(&z));

    vector_delete(&x);
    vector_delete(&y);
    vector_delete(&z);
    return 0;
}
```