課題1

課題 1-A

作成したプログラム

作成したプログラムは下記のようになった

```
/*
A : データ数がすでに分かっている場合にデータとしてリスト構造を持たない配列を使用した場合

a. データの入力順に総和を求めたとき
b. データ入力後に、絶対値に関して昇順に並べ替えた後に総和を求めたとき
*/

#include <stdio.h>

double a(double *inputs, int max_content)
{
        double answer = 0;
        for (int j = 0; j < max_content; j++) {
                answer += inputs[j];
        }
        return answer;
}

double b(double *inputs, int max_content)
{
        double tmp = 0;
        double num_i = 0;
        double num_j = 0;
        double answer = 0;
        for (int i = 0; i < max_content; i++) {
                for (int j = 0; j < max_content; j++) {
                        num_i = inputs[i];
                        if (inputs[i] < 0) {
                                num_i = num_i * -1;
                        }
                        num_j = inputs[j];
                        if (inputs[j] < 0) {
                                num_j = num_j * -1;
                        }
                        if (num_i < num_j) {
                                tmp = inputs[i];
                                inputs[i] = inputs[j];
                                inputs[j] = tmp;
                        }
                }
        }
        for (int j = 0; j < max_content; j++) {
```

```
#ifdef DEBUG
                printf("inputs[%d] = %f\n", j, inputs[j]);
#endif
                answer += inputs[j];
        }
        return answer;
}

int main(void)
{
        char buf[128];
        double input;
        double inputs[128];
        int i;

        i = 0;

        while (fgets(buf, sizeof(buf), stdin) != NULL) {
                sscanf(buf, "%le", &input);
                inputs[i] = input;
#ifdef DEBUG
                printf("inputs[%d] = %f\n", i, input);
#endif
                i++;
        }
        printf("a() = %f\n", a(inputs, i));
        printf("b() = %f\n", b(inputs, i));
        return 0;
}
```

課題 1-B

作成したプログラム

作成したプログラムは下記のようになった

```
/*
B : データ数が未知の場合にデータとしてポインタによる線形リスト構造を使用した場合

a. データの入力順に総和を求めたとき
b. データ入力後に、絶対値に関して昇順に並べ替えた後に総和を求めたとき
*/

#include <stdio.h>
#include <float.h>
#include <stdlib.h>

struct node {
        double content;
        struct node *next;
```

```
};

struct node *insert_rear(double num, struct node *nodes)
{
        struct node *item;
        struct node *top;

        top = nodes;

        item = (struct node *)malloc(sizeof(struct node));
        item->content = num;
        item->next = NULL;

        if (nodes == NULL) {
                nodes = item;
                return nodes;
        }

        while (nodes->next != NULL) {
                nodes = nodes->next;
        }

        nodes->next = item;

        return top;

}

double xabs(double num)
{
        if (num < 0) {
                num = num * -1;
        }
        return num;
}

double get_con(struct node *nodes, int num)
{
        struct node *temp;
        temp = nodes;
        for (int j = 0; j != num; j++) {
                temp = temp->next;
        }
        return temp->content;
}

void swap_con(struct node *nodes, int num1, int num2)
{
        struct node *temp1;
        struct node *temp2;
        double tmp;

        temp1 = nodes;
        temp2 = nodes;
```

```
        for (int i = 0; i != num1; i++) {
                temp1 = temp1->next;
        }
        for (int i = 0; i != num2; i++) {
                temp2 = temp2->next;
        }
        tmp = temp1->content;
        temp1->content = temp2->content;
        temp2->content = tmp;

}

void sort(struct node *nodes, int max_con)
{
        double num_i;
        double num_j;
        for (int j = 0; j < max_con; j++) {
                for (int i = 0; i < max_con; i++) {
                        num_i = get_con(nodes, i);
                        if (num_i < 0) {
                                num_i = num_i * -1;
                        }
                        num_j = get_con(nodes, j);
                        if (num_j < 0) {
                                num_j = num_j * -1;
                        }
                        if (num_i > num_j) {
                                swap_con(nodes, i, j);
                        }
                }
        }
}

double a(struct node *nodes)
{
        double answer = 0;
        struct node *temp = nodes;
        while (temp != NULL) {
                answer += temp->content;
                temp = temp->next;
        }
        return answer;
}

double b(struct node *nodes)
{
        double answer = 0;
        sort(nodes, 20);
        for (int i = 0; i <= 20; i++) {
                answer += get_con(nodes, i);
#ifdef DEBUG
                printf("num[%d] = %f\n", i, get_con(nodes, i));
#endif
```

```
        }
        return answer;
}

int main(void)
{
        char buf[128];
        double input;
        struct node *list;
        int i;

        i = 0;
        list = NULL;

        while (fgets(buf, sizeof(buf), stdin) != NULL) {
                sscanf(buf, "%le", &input);
                list = insert_rear(input, list);
                i++;
        }
        printf("a() = %f\n", a(list));
        printf("b() = %f\n", b(list));
        return 0;
}
```

実行結果

課題A,Bともに入力例は下記の通りである

```
1.0e16
-1.0e2
23
-6.4
3.6e2
-0.01
8.0
-70
5.0e3
1.2e-2
-3.0e3
46
-1.7e3
10
-5.0e2
7.0
-2.0e-3
0.3
-30
3.1
-1.0e16
```

上記の入力例を1_input.txtとして保存している

課題 1-A

実行結果は下記のとおりである

```
$ gcc -Wall 1_A.c
$ ./a.out < 1_input.txt
a() = 54.000000
b() = 52.000000
```

課題 1-B

実行結果は下記のとおりである

```
$ gcc -Wall 1_B.c
$ ./a.out < 1_input.txt
a() = 54.000000
b() = 52.000000
```

課題2

作成したプログラム

作成したプログラムは下記のようになった

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
struct list {
        char name[32];
        int point;
        struct list *next;
};

struct list *ins_char(char *name, int point, struct list *lists)
{
        struct list *head;
        struct list *item;
        struct list *temp;

        item = (struct list *)malloc(sizeof(struct list));
        strcpy(item->name, name);
        item->point = point;
        temp = lists;
        if (temp == NULL) {
```

```
                item->next = NULL;
                return item;
        } else if (temp->next == NULL) {
                if (strcmp(item->name, temp->name) < 0) {
                        item->next = temp;
                        head = item;
                        return head;
                } else {
                        temp->next = item;
                        item->next = NULL;
                        return temp;
                }
        } else {
                head = temp;
                while (temp != NULL) {  // `temp-.nuxt != NULL`で回す
                        while (true) {
                                //先頭との比較
                                //temp->nextとの比較
                                //次に回す
                                if (head == temp
                                    && (strcmp(item->name, temp->name) <= 0)) {
                                        item->next = temp;
                                        head = item;
                                        return head;
                                } else if (temp->next == NULL
                                           ||
                                           ((strcmp(temp->name, item->name) <=
                                             0)
                                            &&
                                            (strcmp
                                             (item->name,
                                              temp->next->name) <= 0))) {
                                        item->next = temp->next;
                                        temp->next = item;
                                        return head;
                                }
                                temp = temp->next;
                        }
                }
        }
        return head;
}

struct list *ins_point(struct list *inputs, struct list *outputs)
{
        struct list *temp;
        struct list *head;
        struct list *item;
        temp = outputs;

        item = (struct list *)malloc(sizeof(struct list));
        strcpy(item->name, inputs->name);
        item->point = inputs->point;
```

```
        if (temp == NULL) {     //0つのとき
                item->next = NULL;
                return item;
        } else if (temp->next == NULL) {        //1つのとき
                if (item->point >= temp->point) {
                        item->next = temp;
                        head = item;
                        return head;
                } else {
                        temp->next = item;
                        item->next = NULL;
                        return temp;
                }
        } else {                //2つのとき
                head = temp;
                //先頭の時
                //temp->next == NULLのとき || temp->nextとの比較
                //次に回す
                while (temp != NULL) {
                        while (true) {
                                if (head == temp
                                    && (item->point >= temp->point)) {
                                        item->next = temp;
                                        head = item;
                                        return head;
                                } else if (temp->next == NULL
                                           || ((temp->point >= item->point)
                                               && (item->point >=
                                                   temp->next->point))) {
                                        item->next = temp->next;
                                        temp->next = item;
                                        return head;
                                }
                                temp = temp->next;
                        }
                }
        }
        return head;
}

void plists(struct list *lists)
{
        struct list *tmp = lists;
        while (tmp != NULL) {
                printf("name = %s, point = %d\n", tmp->name, tmp->point);
                tmp = tmp->next;
        }
}

int main(void)
{
        char buf[128];
        char name[32];
        int point;
```

```
        struct list *lists;
        struct list *point_sorted;
        int i;

        i = 0;
        lists = NULL;

        while (fgets(buf, sizeof(buf), stdin) != NULL) {
                sscanf(buf, "%s %d", name, &point);
                lists = ins_char(name, point, lists);
                i++;
        }
        printf("辞書順によるソート\n");
        plists(lists);
        point_sorted = NULL;
        while (lists != NULL) {
                point_sorted = ins_point(lists, point_sorted);
                lists = lists->next;
        }
        printf("数字順によるソート\n");
        plists(point_sorted);
        return 0;
}
```

実行結果

入力例は下記の通りである

```
DDDD 108
DDCC 107
DDBB 106
DDAA 105
BBBB 104
BBBB 103
BBBB 102
AABB 101
AAAA 100
DDDD 100
```

実行結果は次のようになった

```
$ gcc -Wall 2.c
$ ./a.out < 2_input.txt
辞書順によるソート
name = AAAA, point = 100
name = AABB, point = 101
name = BBBB, point = 102
```

```
name = BBBB, point = 103
name = BBBB, point = 104
name = DDAA, point = 105
name = DDBB, point = 106
name = DDCC, point = 107
name = DDDD, point = 100
name = DDDD, point = 108
数字順によるソート
name = DDDD, point = 108
name = DDCC, point = 107
name = DDBB, point = 106
name = DDAA, point = 105
name = BBBB, point = 104
name = BBBB, point = 103
name = BBBB, point = 102
name = AABB, point = 101
name = DDDD, point = 100
name = AAAA, point = 100
```