

プログラミング演習 第4課題

滝本 亘 (学籍番号 1029-33-1175)

2022 年 7 月 18 日

問題 1(構造体)

作成したプログラムをコード 1 に示す。

コード 1: 構造体

```
1  #include <stdio.h>
2
3  //構造体を宣言
4  typedef struct
5  {
6      int number;
7      int japanese;
8      int math;
9  } student;
10
11 int main (void)
12 {
13     //構造体変数を宣言
14     student std[] =
15     {
16         {1,95,78},
17         {2,45,83},
18         {3,100,90},
19         {4,20,15},
20         {5,43,70},
21     };
22
23     //入力した学籍番号に対応して結果を出力
24     int no;
25     printf("学籍番号を入力してください:");
26     scanf("%d", &no);
27     printf("国語%点、数学 d%点 d", std[no-1].japanese, std[no-1].math);
28 }
```

問題 2(ポインタ)

作成したプログラムをコード 2 に示す。

コード 2: ポインタ

```
1 #include <stdio.h>
2 #define N 2
3
4 int main (void)
5 {
6     //二次元配列を定義
7     int a[N][N] = {{1,2},{3,4}};
8     int b[N][N] = {{5,6},{7,8}};
9     int c[N][N];
10
11     //ポインタを定義
12     int *x, *y, *z;
13     x = &a[0][0];
14     y = &b[0][0];
15     z = &c[0][0];
16
17     //行列の積
18     for(int i=0; i<N; i++){
19         for (int j=0; j<N; j++){
20             *(z+N*i+j) = 0;
21             for(int k=0; k<N; k++){
22                 *(z+N*i+j) += *(x+N*i+k) * *(y+N*k+j);
23             }
24         }
25     }
26
27     //結果を出力
28     for(int i=0; i<N; i++){
29         for(int j=0; j<N; j++){
30             printf("%d_", *(z+N*i+j));
31         }
32         printf("\n");
33     }
34 }
```

結果は、 $c = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$

問題3(動的メモリ確保)

作成したプログラムをコード3に示す。

コード 3: 動的メモリ確保

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define N 1000000
4
5 //配列の順序をランダムに並び替える関数を定義
6 void shuffle(int array[], int size)
7 {
8     for(int i=0; i<size; i++){
9         int j = rand()%size;
10
11         int tmp = array[i];
12         array[i] = array[j];
13         array[j] = tmp;
14     }
15 }
16
17 int main(void)
18 {
19     //百万個のメモリを確保
20     int *x;
21     x=(int*)malloc(N*sizeof(int));
22
23     //メモリに値を代入
24     for(int i=0; i<N; i++){
25         x[i] = i+1;
26     }
27
28     //メモリの値をランダムに並び替え
29     shuffle(x, N);
30
31     //結果を出力
32     for(int i=0; i<N; i++){
33         printf("%d_", x[i]);
34     }
35
36     //メモリの解放
37     free(x);
38 }
```

問題 4(マージソート)

作成したプログラムをコード 4 に示す。

コード 4: マージソート

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define N 128
4
5  //配列の順序をランダムに並び替える関数
6  void shuffle(int array[], int size)
7  {
8      for(int i=0; i<size; i++){
9          int j = rand()%size;
10
11          int tmp = array[i];
12          array[i] = array[j];
13          array[j] = tmp;
14      }
15  }
16
17 //マージソートを実行する関数
18 void merge_sort(int array[], int left, int right)
19 {
20     int i,j,k,mid;
21     int array2[N];
22     int *x,*y;
23
24     x = &array[0];
25     y = &array2[0];
26
27     if(left<right){
28
29         //前半と後半にデータを分け小さい順にソート
30         mid = (left+right)/2;
31         merge_sort(array, left, mid);
32         merge_sort(array, mid+1, right);
33
34         //ソートされた前半の配列をメモリに代入
35         for(i=mid; i>=left; i--){
36             *(y+i) = *(x+i);
37         }
38
39         //ソートされた後半の配列をメモリに代入
40         for(j=mid+1; j<=right; j++){
41             *(y+right-(j-(mid+1))) = *(x+j);
42         }
43
44         i = left;
45         j = right;
46
47         //前半の先頭の値と後半の先頭の値のうち、小さい方を順に取得
48         for(k=left; k<=right; k++){
49             if(*(y+i)<*(y+j)){
50                 *(x+k) = *(y+i++);
51             }
52             else{
53                 *(x+k) = *(y+j--);
54             }
55         }
56     }
57 }
58
59 int main(void)
60 {
61     //配列を定義
62     int a[N];
```

```
63
64     for(int i=0; i<N; i++){
65         a[i] = i+1;
66     }
67
68     //ランダムに並び替え
69     shuffle(a, N);
70
71     //並び替えた後の配列を出力
72     for(int i=0; i<N; i++){
73         printf("%d_", a[i]);
74     }
75
76     printf("\n");
77
78     //マージソートを実行
79     merge_sort(a, 0, N-1);
80
81     //結果を出力
82     for(int i=0; i<N; i++){
83         printf("%d_", a[i]);
84     }
85 }
```

問題5(ナップザック問題)

作成したプログラムをコード5に示す。

コード 5: ナップザック問題

```
1  #include <stdio.h>
2  #include<string.h>
3
4  #define N 16
5  #define C 1800
6
7
8  //構造体を宣言
9  typedef struct
10 {
11     int number;
12     int price;
13     int value;
14 } knap;
15
16 //構造体変数を宣言
17 knap knaps[N] = {
18     {1,178,20},
19     {2,255,24},
20     {3,164,10},
21     {4,334,35},
22     {5,1000,100},
23     {6,108,13},
24     {7,789,70},
25     {8,435,35},
26     {9,95,22},
27     {10,324,21},
28     {11,52,6},
29     {12,248,27},
30     {13,512,60},
31     {14,400,43},
32     {15,122,10},
33     {16,870,87}
34 };
35
36 //ナップザック問題を解く関数
37 int knapsack(int i, int c)
38 {
39     int put_value;
40     int no_put_value;
41     int max_value;
42     int a[N];
43
44     //入れる商品がない場合
45     if (i > N - 1){
46         return 0;
47     }
48
49     //対象の商品を入れない場合の最大価値
50     no_put_value = knapsack(i+1, c);
51
52     //対象の商品が入る場合/
53     if (knaps[i].price <= c) {
54         //対象の商品を入れた場合の価値
55         put_value = knapsack(i+1, c - knaps[i].price) + knaps[i].value;
56     }
57
58     //対象の商品が入らない場合
59     else {
60         //対象の商品を入れない場合の最大価値より小さいため負の値を置く
61         put_value = -1;
62     }
```

```

63
64 //対象の商品を入れた場合と入れなかった場合の大きい方の価値を取得 t
65 if (no_put_value > put_value) {
66     max_value = no_put_value;
67     a[i]=0;
68 }
69
70 else {
71     max_value = put_value;
72     a[i]=1;
73 }
74
75 return max_value;
76 }
77
78 int main(void)
79 {
80     int i;
81     int ret;
82
83     //ナップザック問題の解を取得
84     ret = knapsack(0, C);
85
86     int price = 0;
87
88     //商品の内訳を出力
89     printf("商品の内訳は\n");
90     for(i=0;i<N;i++)
91     {
92         if (knapsack(0,C)<=knapsack(0,C-knaps[i].price)+knaps[i].value){
93
94             price += knaps[i].price;
95             printf("%d,%d,%d\n", knaps[i].number, knaps[i].price, knaps[i].value);
96         }
97     }
98
99     //結果を出力
100    printf("価値の合計は%d\支払額は n%円 d", ret, price);
101
102    return 0;
103 }

```

商品の内訳は

1,178,20

2,255,24

6,108,13

9,95,22

12,248,27

13,512,60

14,400,43

価値の合計は 209

支払額は 1796 円

問題 6(リダイレクト)

作成したプログラムをコード 6 に示す。

コード 6: リダイレクト

```
1 #include<stdio.h>
2
3 int main()
4 {
5     char word[256];
6
7     //文字列の読み込み
8     scanf("%s", word);
9
10    //読み込んだ文字列を出力
11    printf("%s\n",word);
12
13    return (0);
14 }
```

問題 7(モード)

表 1 に示す。

表 1: ファイルのモード

モード	種類・機能	ファイルがないとき
r	テキストファイル、読み取り	エラー
w	テキストファイル、書き込み	新規作成
a	テキストファイル、書き込み (追加)	新規作成
rb	バイナリファイル、読み取り	エラー
wb	バイナリファイル、書き込み	新規作成
ab	バイナリファイル、書き込み (追加)	新規作成
r+	テキストファイル、読み取り、書き込み	エラー
w+	テキストファイル、読み取り、書き込み	新規作成
a+	テキストファイル、読み取り、書き込み (追加)	新規作成
rb+ または r+b	バイナリファイル、読み取り、書き込み	エラー
rb+ または r+b	バイナリファイル、読み取り、書き込み	新規作成
rb+ または r+b	バイナリファイル、読み取り、書き込み (追加)	新規作成

問題 8(関数の使い方)

表 2 に示す。

表 2: 関数の使い方

関数	引数	機能
fprintf	ファイルポインタ, 書き込み文字列, 変数	文字列をファイルに書き込む
fscanf	ファイルポインタ, 書き込み文字列, 変数	文字列をファイルから読み取る
fread	データ, データのバイト長, データの数, ファイルポインタ	バイナリデータの読み込み
fwrite	データ, データのバイト長, データの数, ファイルポインタ	バイナリデータの書き込み

問題 9(fprintf、 fscanf)

作成したプログラムをコード 7 に示す。

コード 7: fprintf、 fscanf

```
1  #include<stdio.h>
2
3  int main()
4  #include<stdio.h>
5
6  int main()
7  {
8      FILE *fp;
9      char x[64];
10
11     //ファイルを書き込みモードで開く
12     fp=fopen("9.txt","w");
13
14     //文字列をファイルに書き込み
15     fprintf(fp,"%s","HelloWorld!");
16
17     //ファイルを閉じる
18     fclose(fp);
19
20     //ファイルを読み取りモードで開く
21     fp=fopen("9.txt","r");
22
23     //文字列をファイルから読み取る
24     fscanf(fp,"%s",x);
25
26     //読み取った文字列を出力
27     printf("%s",x);
28
29     //ファイルを閉じる
30     fclose(fp);
31
32     return 0;
33 }
```

問題 10(fread、fwrite)

作成したプログラムをコード 8 に示す。

コード 8: fread、fwrite ← a

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     //配列を定義
6     int a[5]={1,2,3,4,5};
7     int c[5];
8
9     FILE *fp;
10
11     //ファイルを書き込みモードで開く
12     fp=fopen("10-1.txt","w");
13
14     //ファイルに文字列を書き込み
15     fwrite(a,sizeof(int),5,fp);
16
17     //ファイルを閉じる
18     fclose(fp);
19
20     //ファイルを書き込みモードで開く
21     fp=fopen("10-1.txt","r");
22
23     //文字列をファイルから読み取り
24     fread(c,sizeof(int),5,fp);
25     c[5]='\0';
26
27     //読み取った文字列を出力
28     printf("読み込んだ文字は\n");
29     for(int i=0;i<5;i++){
30         printf("%d",c[i]);
31     }
32
33     //ファイルを閉じる
34     fclose(fp);
35
36     return 0;
37 }
```

作成したプログラムをコード 9 に示す。

コード 9: fread、fwrite - b

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     //配列を定義
6     double b[5]={0.1,0.2,0.3,0.4,0.5};
7     double c[5];
8
9     FILE *fp;
10
11     //ファイルを書き込みモードで開く
12     fp=fopen("10-2.txt","w");
13
14     //文字列をファイルに書き込み
15     fwrite(b,sizeof(double),5,fp);
16
17     //ファイルを閉じる
18     fclose(fp);
19
20     //ファイルを書き込みモードで開く
21     fp=fopen("40-2.txt","r");
22
23     //文字列をファイルから読み取り
24     fread(c,sizeof(double),5,fp);
25     c[5]='\0';
26
27     //読み取った文字列を出力
28     printf("読み込んだ文字は\n");
29     for(int i=0;i<5;i++){
30         printf("%f",c[i]);
31     }
32
33     //ファイルを閉じる
34     fclose(fp);
35
36     return 0;
37 }
```
