

1 2016 年 情報科学 1 期末試験

1.1 問題 1

(プログラムコード: fp-toil.c)

```
1 //解答 (1)2 (2)1 (3)6 (4)6 (5)1 (6)4 (7)4
2
3 #include<stdio.h>
4
5 int main()
6 {
7     int a=20, b=123;
8     double e=10.0,f=3.14;
9     char g='a'; //ASCII コード表を参照 16 進数で 61, 10 進数に変換すると 97
10
11     printf("91:%3d%3d\n", a, b);
12     printf("101:%3d\n", a % b);
13     printf("111:%5.2f%5.2f\n", e,f); //5 桁の小数点第 2 位
14     printf("121:%d\n", g-3); // 10 進数 97-3=94
15     printf("131:%c\n", g-3); // ASCII コード表を参照 文字 a-3=^
16     printf("141:%x\n", g-3); // 16 進数 61-3=5e %X(大文字) なら 5E
17     printf("151:%p\n",&a);
18     printf("161:%p\n",&a+1); // int は 4 バイトのメモリ領域をとるので、15 行目で出力したアド
    レスから 4 増えたアドレスとなる
19
20     return 0;
21 }
```

1.2 問題 2

(プログラムコード: fp-toi2.c)

```
1 //解答 (1)00 (2)06 (3)4 (4)2 (5)3
2
3 #include<stdio.h>
4
5 int main()
6 {
7     int i;
8
9     for(i = 5; i < 17; i += 2){ //i の値は 5 から始まり 7,9,11,13,15 と 2 ずつ加算される
10         if(i ==8){
11             break; //i の値が 8 になることはないため break はしない
12         }
13         printf("i=%d\n",i);
14     }
15
16     return 0;
17 }
```

1.3 問題3

(プログラムコード: fp-toi3.c)

```
1  //解答 (1)1 (2)0 (3)1 (4)0 (5)8 (6)4 (7)0 (8)5 (9)3 (10)3 (11)1
2
3  #include<stdio.h>
4
5  int Swap1(int a, int b);
6  int Swap2(int *a, int *b);
7
8  int main()
9  {
10     int a = 3, b = 4, c = 5, d = 9;
11
12     c = Swap1(a, b);
13     printf("a=%d, b=%d, c=%d, d=%d\n", a, b, c, d);
14
15     d = Swap2(&a, &b);
16     printf("a=%d, b=%d, c=%d, d=%d\n", a, b, c, d);
17
18     return 0;
19 }
20
21 int Swap1(int a, int b){
22     int tmp;
23     tmp = a;
24     a = b;
25     b = tmp;
26     return tmp;
27 }
28
29 int Swap2(int *a, int *b){
30     int tmp;
31     tmp = *a;
32     *a = *b;
33     *b = tmp;
34     return tmp;
35 }
```

1.4 問題 4

(プログラムコード: fp-toi4.c)

```
1  //解答 (1)0 (2)5 (3)3 (4)6 (5)7 (6)2 (7)2 (8)3 (9)0 (10)3 (11)4
2
3  #include<stdio.h>
4  #include<string.h>
5
6  struct PhonesTag{
7      char name[32];
8      char tel[32];
9      int gid;
10 };
11
12 int InputPhones(struct PhonesTag *pPhone);
13 void printPhones(int num, struct PhonesTag *pPhone);
14
15 int main(){
16     struct PhonesTag phones[32];
17     int num, ret;
18
19     for(num = 0; num < 32; num++){
20         ret = InputPhones(&phones[num]);
21         if(ret != 0){
22             break;
23         }
24     }
25
26     printPhones(num, phones);
27
28     return 0;
29 }
30
31 int InputPhones(struct PhonesTag *pPhone){
32     printf("Input your name:");
33     scanf(" %[^\n]", pPhone->name);
34     if(strcmp(pPhone->name, "end")==0){
35         return -1;
36     }
37
38     printf("Input your phone number:");
39     scanf(" %[^\n]", pPhone->tel);
40 }
```

```
41     printf("Input group id:");
42     scanf(" %d", &(pPhone->gid));
43     return 0;
44 }
45
46 void printPhones(int num, struct PhonesTag *pPhone){
47     int i;
48     for(i = 0; i < num; i++){
49         printf("**** member %d ****\n", i);
50         printf("name : %s\n", pPhone[i].name);
51         printf("phone : %s\n", pPhone[i].tel);
52         printf("group id : %d\n", pPhone[i].gid);
53     }
54     return;
55 }
```

1.5 問題 6

(プログラムコード: fp-toi6.c)

```
1  //解答 (1)0 (2)1 (3)1 (4)1 (5)2 (6)1 (7)3
2
3  #include<stdio.h>
4
5  struct Stack {
6      int data[100];
7      int stackPointer;
8  };
9
10 void push(struct Stack *st, int n);
11 int pop(struct Stack *st);
12
13 int main()
14 {
15     int i, n;
16     struct Stack st;
17
18     // スタックと stackPointer の初期化
19     st.stackPointer = 0;
20     for(i = 0; i < 100; i++){
21         st.data[i] = 0;
22     }
23
24     push(&st, 25);
25     push(&st, 32);
26     n = pop(&st);
27     push(&st, 55);
28
29     n = pop(&st);
30     printf("%d\n", n);
31     n = pop(&st);
32     printf("%d\n", n);
33     n = pop(&st);
34     printf("%d\n", n);
35
36     for(i = 0; i < 10; i++){
37         push(&st, i);
38     }
39
40     push(&st, 10);
```

```
41
42     return 0;
43 }
44
45
46 void push(struct Stack *st, int n){
47
48     if(st->stackPointer == 100){
49         printf("Stack is Full.\n");
50         //途中で終わらせるため
51         return;
52     }
53
54     st->data[st->stackPointer] = n;
55     st->stackPointer++;
56 }
57
58
59 int pop(struct Stack *st){
60     if(st->stackPointer == 0){
61         printf("Stack is empty\n");
62         return -1;
63     }
64
65     st->stackPointer--;
66     return st->data[st->stackPointer];
67 }
```
