

# H29 年度 J3 データ構造アルゴリズム 前期中間試験問題

2017 年 6 月 6 日 (火) 11 時 20 分～12 時 20 分 (60 分)

## ■ 受験時の注意 ■

- ・問題全体を通して、「¥」 は「バックスラッシュ」に置き換えて考えてくださいね！
- ・プログラムを短くするために、例えば、malloc()によるメモリ確保時の確保できたかのチェックや、解放時の free()など、本来行うべき処理を色々と省いています！けど、許してー！
- ・プログラムは実行して確認しています！セキュアな例外処理は皆無ですので、耐えられない入力が多々あります！しかしながら、この問題における入力では動きます！ので、許してー！
- ・解答欄には必ず何か書いておいてください！できる限り部分点をつけますよー！
- ・恐らく難易度の感じ方は皆さんそれぞれだと思います！ので、さっと見て、ちょっと時間かかるかなって思ったらどんどん次の問題に取り組んでください！
- ・意地悪な作りはしてないつもりです！ので、なんとなく読んで、大枠どんなことをしているプログラムだなって分かれば答えられるはずですよ！ってなつもりで作ってます！ので、勘で答えてもらっても良いですよ！なんとなく分かるってことは、分かるってことです！
- ・あとは、持ち込みルールなど、授業中に説明しましたが、もう一度書いておきますね！
  - 全て持ち込み可！
  - ノート PC、携帯などにて調べることも可！
  - プログラムを実際にコンパイルして動かすと結果が出てくるので、それは無し！
  - 相談も無し！
  - これらに類する行為も無し！
  - クラスのみんなにフェアに楽しみましょう！

以下のプログラムの実行結果を答えましょう！(1)～(20) 全て 5 点、なので合計 100 点です！

(1)

```
#include <stdio.h>
```

```
int main(void)
{
    int i = 5;

    printf("%d¥n", i);

    return 0;
}
```

(2)

```
#include <stdio.h>
```

```
int main(void)
{
    int i = 5;
    int *ip = &i;
```

```

        printf("%d\\n", *ip);

        return 0;
}

```

(3)

```

#include <stdio.h>

int main(void)
{
    int a[5] = {0, 1, 2, 3, 4};
    int ans = 0;
    int i;

    for ( i = 0; i < 5; i++ ) {
        ans += a[i];
    }
    printf("%d\\n", ans);

    return 0;
}

```

(4)

```

#include <stdio.h>

int main(void)
{
    char str[15] = "HelloWorld";
    char *cp;

    cp = &str[5];
    printf("%s\\n", cp);

    return 0;
}

```

(5)

```

#include <stdio.h>

typedef struct Point {
    int x;
    int y;
} Point;

int main(void)
{
    Point pt;
    Point *p;

    pt.x = 5;
    pt.y = 10;
    p = &pt;
    printf("%d, %d\\n", p->x, pt.y);

    return 0;
}

```

(6)

```
#include <stdio.h>

int main(void)
{
    int a[5] = {10, 20, 30, 40, 50};
    int i, index;

    for ( i = 0; i < 5; i++ ) {
        if ( a[i] == 30 ) {
            index = i;
            break;
        }
    }
    printf("%d\n", index);

    return 0;
}
```

(7)

```
#include <stdio.h>

void f(int *x, int *y);

int main(void)
{
    int x = 3, y = 5;

    f(&x, &y);
    printf("%d, %d\n", x, y);

    return 0;
}

void f(int *x, int *y)
{
    int a = *x;
    *x = *y;
    *y = a;
}
```

(8)

```
#include <stdio.h>

int f(char str[]);

int main(void)
{
    char str[] = "Hello";

    printf("%d\n", f(str));

    return 0;
}

int f(char str[])
{
    int i;
```

```

        for ( i = 0; str[i] != '\0'; i++ );

        return i;
}

```

(9)

```
#include <stdio.h>
```

```
void f(char str1[], char str2[]);
```

```
int main(void)
```

```
{
    char str1[] = "Hello";
    char str2[10];

    f(str1, str2);
    printf("%s\n", str2);

    return 0;
}
```

```
void f(char str1[], char str2[])
```

```
{
    int i;

    for ( i = 0; str1[i] != '\0'; i++ ) {
        str2[i] = str1[i];
    }
    str2[i] = '\0';
}
```

(10)

```
#include <stdio.h>
```

```
void f(char str1[], char str2[]);
```

```
int main(void)
```

```
{
    char str1[20] = "Hello";
    char str2[] = "World";

    f(str1, str2);
    printf("%s\n", str1);

    return 0;
}
```

```
void f(char str1[], char str2[])
```

```
{
    int i, j;

    for ( i = 0; str1[i] != '\0'; i++);
    for ( j = 0; str2[j] != '\0'; j++ ) {
        str1[i + j] = str2[j];
    }
    str1[i + j] = '\0';
}
```

```

(11)
#include <stdio.h>

void f(int a[2][3]);

int main(void)
{
    int a[2][3];
    int i, j;

    for ( i = 0; i < 2; i++ ) {
        for (j = 0; j < 3; j++ ) {
            a[i][j] = i + j;
        }
    }
    f(a);

    return 0;
}

void f(int a[2][3])
{
    int i, j;

    for ( i = 0; i < 2; i++ ) {
        for (j = 0; j < 3; j++ ) {
            printf("%d", a[i][j]);
        }
        printf("\n");
    }
}

```

```

(12)
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    char *str;

    str = (char *)malloc(sizeof(char) * 5);

    str[0] = 'H';
    str[1] = 'e';
    str[2] = '\0';

    printf("%s\n", str);

    return 0;
}

```

```

(13)
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int *ip;
    int i;
    int ans = 0;

    ip = (int *)malloc(sizeof(int) * 5);

    for ( i = 0; i < 5; i++ ) {
        ip[i] = i;
    }

    for ( i = 0; i < 5; i++ ) {
        ans += ip[i];
    }

    printf("%d\n", ans);

    return 0;
}

```

```

(14)
#include <stdio.h>
#include <stdlib.h>

typedef struct CELL {
    int value;
    struct CELL *next;
} CELL;

int main(void)
{
    CELL *p;

    p = (CELL *)malloc(sizeof(CELL));
    p->value = 5;
    p->next = NULL;
    printf("%d\n", p->value);

    return 0;
}

```

```

(15)
#include <stdio.h>
#include <stdlib.h>

typedef struct CELL {
    int value;
    struct CELL *next;
} CELL;

```

```

int main(void)
{
    CELL root;
    CELL *p = &root;
    int a[3] = {3, 5, 7};
    int i;

    for ( i = 0; i < 3; i++ ) {
        p->next = (CELL *)malloc(sizeof(CELL));
        p = p->next;
        p->value = a[i];
    }
    p->next = NULL;

    for (p = root.next; p != NULL; p = p->next ) {
        printf("%d\\n", p->value);
    }

    return 0;
}

```

(16)

```

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

typedef struct CELL {
    int value;
    struct CELL *next;
} CELL;

int main(void)
{
    CELL root;
    CELL *p = &root;
    CELL *x;
    int a[3] = {3, 5, 7};
    int i;

    for ( i = 0; i < 3; i++ ) {
        p->next = (CELL *)malloc(sizeof(CELL));
        p = p->next;
        p->value = a[i];
    }
    p->next = NULL;

    p = root.next;
    x = &root;
    x->next = (CELL *)malloc(sizeof(CELL));
    x = x->next;
    x->value = 1;
    x->next = p;

    for (p = root.next; p != NULL; p = p->next ) {
        printf("%d\\n", p->value);
    }

    return 0;
}

```

```

(17)
#include <stdio.h>
#include <stdlib.h>

typedef struct CELL {
    int value;
    struct CELL *next;
} CELL;

void f(CELL *p);

int main(void)
{
    CELL root;
    CELL *p = &root;
    int a[3] = {3, 5, 7};
    int i;

    for ( i = 0; i < 3; i++ ) {
        p->next = (CELL *)malloc(sizeof(CELL));
        p = p->next;
        p->value = a[i];
    }
    p->next = NULL;

    p = root.next;
    f(p);

    for (p = root.next; p != NULL; p = p->next ) {
        printf("%d\\n", p->value);
    }

    return 0;
}

void f(CELL *p)
{
    p->next = p->next->next;
}

```



```

(18)
#include <stdio.h>
#include <stdlib.h>

typedef struct CELL {
    int value;
    struct CELL *next;
} CELL;

int main(void)
{
    CELL root;
    CELL *p = &root;
    int a[5] = {3, 5, 2, 4, 1};
    int x;
    CELL *i, *j;

    for ( x = 0; x < 5; x++ ) {
        p->next = (CELL *)malloc(sizeof(CELL));
        p = p->next;
        p->value = a[x];
    }
    p->next = NULL;

    for ( i = root.next; i->next != NULL; i = i->next ) {
        for ( j = i->next; j != NULL; j = j->next ) {
            if (i->value > j->value) {
                x = i->value;
                i->value = j->value;
                j->value = x;
            }
        }
    }

    for (p = root.next; p != NULL; p = p->next ) {
        printf("%d\\n", p->value);
    }

    return 0;
}

```

```

(19)
#include <stdio.h>

void enq(int q[10], int *r, int data);
int deq(int q[10], int *f);

```

```

int main(void)
{
    int q[10];
    int f = 0, r = 0;

    enq(q, &r, 5);
    enq(q, &r, 3);
    printf("%d\\n", deq(q, &f));
    printf("%d\\n", deq(q, &f));

    return 0;
}

void enq(int q[10], int *r, int data)
{
    q[(*r)++] = data;
}

int deq(int q[10], int *f)
{
    int x = q[(*f)++];
    return x;
}

(20)
#include <stdio.h>

void push(int s[3], int *sp, int data);
int pop(int s[3], int *sp);

int main(void)
{
    int s[3];
    int sp = 0;

    push(s, &sp, 5);
    push(s, &sp, 3);
    printf("%d\\n", pop(s, &sp));
    printf("%d\\n", pop(s, &sp));

    return 0;
}

void push(int s[3], int *sp, int data)
{
    s[(*sp)++] = data;
}

int pop(int s[3], int *sp)
{
    return s[--*sp];
}

```

以上です！