

1. 別紙のプログラム (p1.c) の実行結果を答えなさい (3 点 × 5 行=15 点)

\$ p1

10

20

10

10

20

2. 別紙のプログラム (p2.c) と実行結果をよく読んで答えなさい。

- (1) mystrcpy() 関数を完成しなさい。(6 点)

```
void mystrcpy(char *p, char *q) {
```

```
    while(*q != '\0') {
```

```
        *p = *q;
```

```
        p++;
```

```
        q++;
```

```
    }
```

```
    *p = '\0';
```

```
}
```

- (2) mystrlen() 関数を完成しなさい。(7 点)

```
int mystrlen(char *p) {
```

```
    char *q = p;
```

```
    while (*p != '\0')
```

```
        p++;
```

```
    return p - q;
```

```
}
```

- (3) mystrcat() 関数を完成しなさい。(7 点)

```
void mystrcat(char *p, char *q) {
```

```
    while (*p != '\0')
```

```
        p++;
```

```
    while (*q != '\0') {
```

```
        *p = *q;
```

```
        p++;
```

```
        q++;
```

```
    }
```

```
    *p = '\0';
```

```
}
```

- (4) myreverse() 関数を完成しなさい。(7 点)

```
void myreverse(char *p) {
```

```
    char *q = p;
```

```
    while (*q != '\0')
```

```
        q++;
```

```
    q--;
```

```
    while(q > p) {
```

```
        char c = *p;
```

```
        *p = *q;
```

```
        *q = c;
```

```
        p++;
```

```
        q--;
```

```
    }
```

```
}
```

3. 別紙のプログラム (p3.c) をよく読んで答えなさい。

- (1) p3f1() 関数は、文字列が ASCII コードで表現される英単語であることを前提に、二つの単語が辞書に出てくる順を判断している。どのような時、ゼロ、正、負の値が返されるか説明しなさい。(5 点)

二つの単語が同じならゼロを返す。
p の単語が辞書で前に出てくるなら負の値を返す。
p の単語が辞書で後に出てくるなら正の値を返す。

- (2) p3f2() 関数は、何をする関数か説明しなさい。(5 点)

str 配列の n 件の文字列について、辞書順にソートする関数である。

- (3) プログラム p3 の実行結果を答えなさい。(5 点)

\$ p3

car
cat
dog
moon
sun

4. 別紙のプログラム (p4.c) の空欄 (???部分) に適切な記述を答えなさい。なお、記述は実行結果と合致するあること。また、記述は構造体または構造体フィールド (メンバ) を単にアクセスするものであり、計算等を含まないこと。(4 点 × 7 問=28 点)

番号	記 述	番号	記 述
(1)	[0].x	(2)	->x
(3)	[1].x	(4)	.n
(5)	.n[1]	(6)	.xy2.x
(7)	xya[1]		

5. 別紙のプログラム (p5.c) をよく読んで答えなさい。

- (1) プログラム p5 の実行結果を答えなさい。(5 点)

\$ p5

1,2
2,1

- (2) プログラム p5-1.c のような main() 関数から呼ばれたとき、p5 と同じ実行結果になる printXY を書きなさい。(5 点)

```
void printXY(struct XY *xy) {  
    printf("%d,%d\n", xy->x, xy->y);  
}
```

- (3) プログラム p5-1.c のような main() 関数から呼ばれたとき、同じ実行結果になる swapXY を書きなさい。但し、swapXY が返す値は引数のデータを反映したものであること。(定数を swapXY 関数中に書いてはならない。)(5 点)

```
void swapXY(struct XY *xy) {  
    int tmp;  
    tmp = xy->x;  
    xy->x = xy->y;  
    xy->y = tmp;  
}
```

```
// p1.c
#include <stdio.h>
#include <stdlib.h>
void p1f1(int x, int y) { x = y; }
void p1f2(int *x, int y) { *x = y; }
void p1f3(int x, int *y) { x = *y; }
void p1f4(int *x, int *y) { x = y; }
void p1f5(int *x, int *y) { *x = *y;}
int main() {
    int a;
    int b = 20;
    a = 10;
    p1f1(a, b);
    printf("%d\n", a);
    a = 10;
    p1f2(&a, b);
    printf("%d\n", a);
    a = 10;
    p1f3(a, &b);
    printf("%d\n", a);
    a = 10;
    p1f4(&a, &b);
    printf("%d\n", a);
    a = 10;
    p1f5(&a, &b);
    printf("%d\n", a);
    return 0;
}
```

```
// p2.c
#include <stdio.h>
#include <stdlib.h>

// 文字列 (q) をバッファ (p) にコピーする
void mystrcpy(char *p, char *q) {
    ... 省略 ...
}

// 文字列 (p) に文字列 (q) を追加する
void mystrcat(char *p, char *q) {
    ... 省略 ...
}

// 文字列の長さを求める
int mystrlen(char *p) {
    ... 省略 ...
}

// 文字列の左右を入れ替える
void myreverse(char *p) {
    ... 省略 ...
}

int main() {
    char buf[100];
    mystrcpy(buf, "123");
    mystrcat(buf, "abc");
    mystrcat(buf, "DEF");
    printf("%s\n", buf);
    printf("%d\n", mystrlen(buf));
    myreverse(buf);
    printf("%s\n", buf);
    return 0;
}

/* 実行結果
$ p2
123abcDEF
9
FEDcba321
*/
```

```
// p3.c
#include <stdio.h>
#include <stdlib.h>

int p3f1(char *p, char *q) {
    while (*p == *q) {
        if (*p=='\0') return 0;
        p++;
        q++;
    }
    return *p - *q;
}

void p3f2(char *str[], int n) {
    for (int i=0; i<n-1; i++) {
        int min = i;
        for (int j=i+1; j<n; j++) {
            if (p3f1(str[j], str[min])<0)
                min = j;
        }
        char *tmp = str[i];
        str[i] = str[min];
        str[min] = tmp;
    }
}

int main() {
    char *str[] = {
        "car", "dog", "cat", "sun", "moon" };
    p3f2(str, 5);
    for (int i=0; i<5; i++)
        printf("%s\n", str[i]);
    return 0;
}
```

```
// p4.c
#include <stdio.h>
#include <stdlib.h>

struct XY {
    int x;
    int y;
};

struct XYXY {
    char *n;
    struct XY xy1;
    struct XY xy2;
};

struct XY xy = {1,2};
struct XY xya[] = {{3,4},{5,6}};
struct XY *xyp;
struct XYXY xyxy= {"ABC",{7,8},{9,10}};

int main() {
    printf("%d\n", xy.x);
    printf("%d\n", xya???);           // (1)
    xyp = &xy;
    printf("%d\n", xyp???);           // (2)
    xyp = xya;
    printf("%d\n", xyp???);           // (3)
    printf("%s\n", xyxy???);          // (4)
    printf("%c\n", xyxy???);          // (5)
    printf("%d\n", xyxy???);          // (6)
    xy = ???;                          // (7)
    printf("%d,%d\n", xy.x, xy.y);
    return 0;
}

/* 実行結果
$ p4
1
3
1
5
B
9
5,6
*/
```

```
// p5.c
#include <stdio.h>
#include <stdlib.h>

struct XY {
    int x;
    int y;
};

struct XY swapXY(struct XY xy) {
    struct XY tmp;
    tmp.x = xy.y;
    tmp.y = xy.x;
    return tmp;
}

void printXY(struct XY xy) {
    printf("%d,%d\n", xy.x, xy.y);
}

int main() {
    struct XY a = {1,2};
    struct XY b;

    printXY(a);
    b = swapXY(a);
    printXY(b);

    return 0;
}
```

```
// p5-1.c
#include <stdio.h>
#include <stdlib.h>

struct XY {
    int x;
    int y;
};

void swapXY(
    ... 省略 ...
)

void printXY(
    ... 省略 ...
)

int main() {
    struct XY a = {1,2};
    struct XY b;

    printXY(&a);
    swapXY(&a);
    printXY(&a);

    return 0;
}
```