

長崎大学大学院生産科学研究科（博士前期課程）
電気情報工学専攻（情報システム工学系）
平成 18 年度入試問題
ソフトウェア

受験番号_____

氏名_____

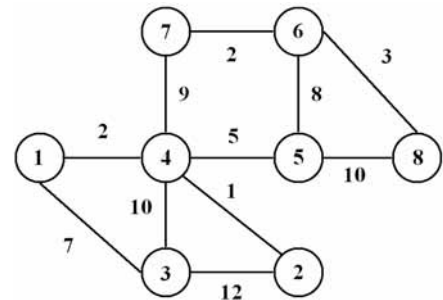
1. 次の問いに答えよ。

- (1) 分割コンパイルとは何かを説明し、その利点を挙げよ。
- (2) アルゴリズムの計算量を表す O 記法を簡単に説明し、 $O(1)$, $O(n)$, $O(\log n)$ で表される 3 つの計算量のうち、どれが優れているかを説明せよ。
- (3) Unix のファイル管理において、マルチユーザ機能のためにどのようなファイルアクセス権限が実現されているかを説明せよ。

〔解答欄〕

2. 図のような無向グラフで与えられる経路において、節 1 を開始点とし、節 8 に到達する最短経路を求める問題において、下記の問に答えよ。但し、節間の線分上の数字は距離値を表わすものとする。

- () この問題をダイクストラのアルゴリズムで解くときの隣接行列を記せ。
- () 処理の順序を時間順に説明するアルゴリズムの表現法を用いて、探索ずみの節が3つ決定されるまでの処理過程（内容）を、隣接行列を用いて説明せよ。
- () 問題を解く過程で、探索ずみとなる節とその推移を距離の推移と共に記せ。また最短経路の節を番号で示せ。



経路のグラフ表現

【解答欄】

() 隣接行列

	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

()

()

3. クイックソートについて、下記の問に答えよ。

- () バブルソートなどの単純ソートと比較した場合のクイックソートの特長を記せ。
- () 下記の配列 DT[0] ~ DT[11] をクイックソートによって昇順にソートせよ。答えはソートの過程を、1 段目の探索結果と 2 段目の探索結果までを数値の並びで示せ。なお、基準値は中央位置である DT[5] とする。
7 7 , 2 4 , 6 8 , 9 0 , 3 1 , 4 6 , 1 2 , 8 , 3 4 , 8 2 , 5 6 , 1 5
- () 安定なソートとはどのようなソートか。また、クイックソートは安定なソートか否か記せ。
- () クイックソートの計算量を比較回数で現すものとし、計算量が 最小の場合と、 最多の場合について求めよ。また、計算量をオーダーで表わすと、 最小の場合、 最多の場合でどのようになるか。さらに、一般的にクイックソートのオーダーはどの値が使われるか。ただし、データ量は n 個、グループ分けの回数を k として示せ。

【解答欄】

()		
()		7 7 , 2 4 , 6 8 , 9 0 , 3 1 , 4 6 , 1 2 , 8 , 3 4 , 8 2 , 5 6 , 1 5
	一段目	
	二段目	
()		安定なソートとは、
		クイックソートは () なソートである。
()		

4. 以下のようにあるファイル名が大域変数 `fname` に定義されているとする。

```
const char fname[] = "/home/luser/bar";
```

このファイル名のファイルに対し、以下に説明する、ファイル情報を返す関数 `stat`、時刻表現形式を変換する関数 `gmtime`、出力関数 `printf` とを使い、その (UTC 標準時で表した) 最終更新時刻を以下の形式 (月日や分秒は表示せず、何時かのみ) で標準出力に出力する C 言語プログラム `chkbar.c` を完成させよ。コンパイル時に型に関する警告やエラーが出ないこと。なお、`/home/luser/bar` がないなどの、実行中に上記のライブラリ関数でエラーが起きる状況に対処する必要はない。

実行例 (出力の形式) `/home/luser/bar` が最後に更新されたのは、ある日の 14 時だとすると：

```
$ ./chkbar
Updated at 14 o'clock.
$
```

\$はプロンプトである。

stat のマニュアル

名前

`stat` - ファイル情報を得る

概要 (プロトタイプ宣言とインクルードすべきもの)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
```

```
int stat(const char *file_name, struct stat *buf);
```

記述

この関数は与えられたファイルに関する情報を返す。あなたはこの情報を得るために、このファイルに関するどのようなアクセス権も持つ必要はないが、そのファイルに向かうパス中出现する全てのディレクトリにおいて検索するアクセス権を持つ必要がある。

`stat` は `file_name` によって指されるファイルの情報を調べ、`buf` をうめる。
すなわち、ファイルの情報全ては、以下のフィールドからなる一つの `stat` 構造体として返される。

```
struct stat {
    dev_t      st_dev;      /* device */
    ino_t      st_ino;      /* inode */
    mode_t     st_mode;     /* protection */
    nlink_t    st_nlink;    /* number of hard links */
    uid_t      st_uid;      /* user ID of owner */
    gid_t      st_gid;      /* group ID of owner */
    dev_t      st_rdev;     /* device type (if inode device) */
    off_t      st_size;     /* total size, in bytes */
    blksize_t  st_blksize;  /* blocksize for filesystem I/O */
    blkcnt_t   st_blocks;   /* number of blocks allocated */
    time_t     st_atime;    /* time of last access */
    time_t     st_mtime;    /* time of last modification(最終更新時刻) */
    time_t     st_ctime;    /* time of last status change */
};
```

フィールド `st_atime` は例えば `execvs(2)`, `mknod(2)`, `pipe(2)`, `utime(2)`, そして (0 バイト以上の) `read(2)` といったファイルアクセス処理によって更新される。 `time_t` 型はカレンダー時刻を表す型である。

返値

エラーがなければ 0 が返される。エラーなら -1 が返される。(エラーは起きないとしてプログラムしてよい)

長崎大学大学院生産科学研究科 (博士前期課程)
電気情報工学専攻 (情報システム工学系)
平成 18 年度 入学試験問題
ソフトウェア

受験番号 _____

氏名 _____

gmtime のマニュアル

名前

gmtime - カレンダー時刻を分解された時間形式に変換する

概要 (プロトタイプ宣言とインクルードすべきもの)

```
#include <time.h>
```

```
struct tm *gmtime(const time_t *timep);
```

記述

この gmtime() 関数はカレンダー時刻を唯一の引数としてとる。
返値である分解された時刻は、以下のように、<time.h>で定義された構造体 tm 型に格納される。

```
struct tm {
    int      tm_sec;          /* seconds(秒) */
    int      tm_min;          /* minutes(分) */
    int      tm_hour;         /* hours(時) */
    int      tm_mday;         /* day of the month(日) */
    int      tm_mon;          /* month(月) */
    int      tm_year;         /* year(年) */
    int      tm_wday;         /* day of the week(曜日) */
    int      tm_yday;         /* day in the year(1月1日からの日数) */
    int      tm_isdst;        /* daylight saving time(サマータイム) */
};
```

この構造体 tm のメンバーの意味は以下の通りである：

tm_sec (分より上の単位で表せなかった残りの) 秒数。0 から 59 の値を取る。
しかしうるう秒では 61 までの値を取るかもしれない。

tm_min (時より上の単位で表せなかった残りの) 分数。0 から 59 の値を取る。

tm_hour 真夜中から測った時間。0 から 23 の値を取る。

tm_mday 月の中での日数。1 から 31 の値を取る。

tm_mon 1 月から数えた月数。0 から 11 の値を取る。

tm_year 1900 から数えた年数。

(途中省略)

この gmtime() 関数はカレンダー時刻 timep を、対応する (UTC 標準時を基準とする) 分解された時間表現に変換する。
もし年数が int に収まらないなら、NULL を返すかもしれない。結果の構造体は静的に割り当てられた領域 (他の日時や時刻関数を引き続き呼出すと上書きされるかもしれない) に置かれ、この関数はそこへのポインタを返す。

返値

上で述べた通りの値を返す。エラーが検出されたら NULL を返す。
(エラーは起きないとしてプログラムしてよい)

長崎大学大学院生産科学研究科 (博士前期課程)
電気情報工学専攻 (情報システム工学系)
平成 18 年度 入学試験問題
ソフトウェア

受験番号 _____

氏名 _____

解答欄 (chkbar.c)

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <time.h>

const char fname[] = "/home/luser/bar";

int
main ()
{
    /* 以下の変数は使わなくてもよい */
    struct stat buf;
    struct tm *gmt;
    /* ここから埋めること。他に必要な変数がある場合もここから定義すること */

    /* この先には追加しないこと */
    return 0;
}
```