

プログラミング言語演習Ⅳ 演習11 氏名： _____ 学籍番号： _____

1. 下線部に適切な語句または数値を記しなさい ただし同じ番号（①②等）を付した部分には同じ語句が入る
- プログラミング上の問題点には大きく分類して、1) _____ 時のエラー、2) _____ 時のエラー、3) 結果が期待通りに出力されない、の3つがある。このうち1)についてはエラーメッセージを丁寧に読むことで対応できる。2,3)の対処法としてはエラーで停止している場所を特定したり、途中まで正常に動作しているかを知るために、適宜printf文を挿入することが考えられる。このとき、出力が直ちに行われるように、_____ をprintfの後に付けておくが良い。また、Unix上のデバグガとして① _____ を用いることもできる。この場合コンパイル時に、gcc source.c _____ -o executable のようにオプションを付けておく必要がある。このプログラムを通常実行する場合、executable a1 a2 のように2つのオプションを取るものであるとすると、まず、① _____ executable としてデバグガを立ち上げ、_____ とした後、_____ と入力すると、エラーが起きて停止したソースファイル上の行番号が表示される。さらに、その時の変数 i の値、を知りたいければ _____ i とすれば、表示される。① _____ の終了時には _____ と入力し、本当にやめるか聞かれるので、_____ と入力する。
2. 以下の様なデータファイルを読み込み、pref_infoという名前の構造体を定義してメモリ上に情報を展開したい この様な目的で定義する構造体について、左側の下線部を埋めなさい

_____ struct pref_info {
_____ pref_name[12];
_____ population;
_____ area;
_____ num_city;
} _____ ;

	人口	面積	市
GUNMA	1,974,555	6,362	12
TOCHIGI	1,979,039	6,408	12
YAMANASHI	838,260	4,465	13

3. プログラム中で int num_pref; という変数を定義し、読み込んだファイルの県の数（上の例では3）が、変数 num_pref に代入されているとする。このとき、ファイルにあるデータを全てメモリ上に展開するための記憶領域を prefs という変数名の構造体配列をmalloc を使って確保するために、必要なコードについて、以下の下線部を埋めなさい
- _____ prefs;

prefs = (_____) malloc (_____ (_____) * (_____));
4. 2. で定義した構造体のサイズは _____ バイトであり、この構造体型の変数prefsとして3で確保した num_pref個の県に関する記憶領域は _____ バイトである。
5. 3.の prefs というポインタ変数の持つアドレス値を、仮に 0（ゼロ）とした場合、以下の各要素の先頭アドレスを整数値で右側の下線部に書きなさい。

prefs[0].pref_name _____ prefs[0].population _____

prefs[0].area _____ prefs[0].num_city _____

prefs[1].pref_name _____ prefs[2].num_city _____

6. ヘッダファイルは、 _____ というサフィックスを持ち、一般に _____ 、 _____ 、 _____ などを記述する。自分で作成したcity. _____ というヘッダファイルの内容を同じディレクトリに在るcity.cの中に反映（展開）させたい場合には、city.c の冒頭で #include _____ のように記述する。

7. 以下の16進数の和を求めなさい。答えも16進数で記述すること

0x04a0 + 8 = _____	0xaaaf0 + 1 = _____
0x04a4 + 8 = _____	0xaaaf9 + 1 = _____
0x04a8 + 8 = _____	0xaaaff + 1 = _____

8. 横に並んだ16進数の間の差分を下の下線部に記しなさい

0x1738	0x173c	0x176c	0x1774	0x1778
_____	_____	_____	_____	_____

9. 以下の下線部に適切な語句を入れなさい

- a) Cで宣言した変数名には変数値が入る場所の他に、一つのポインタが割り当てられていて、宣言した変数名の前に_____記号をつけることでそのアドレス値（変数値のはいる場所の先頭）を参照できる。
- b) 変数宣言時に型名の後（変数名の前）に_____記号をつけると、その型のポインタ変数が定義できる。
- c) アドレス値を表す変数（ポインタ変数）の前に_____記号をつけると、そのアドレスの指し示す場所にあるポインタ変数の型の変数値を参照できる。また、この参照方法で変数値を書き換えることもできる。
- d) intのポインタ変数にnを加えると、_____ 掛けるn だけ、ポインタ変数の値が増える。

10. 整数型の配列 int ai[5]; を定義したとき、2番目の要素の値を参照する書き方を2つ記しなさい。

また、2番目の要素のあるアドレス値を参照する書き方を2つ記しなさい。

11. 以下の様に変数宣言を行ったとき、下に示す値を出力するために記すprintf文の下線部を埋めなさい。ただし、最初に定義される変数のアドレスの初期値としては 0x1000 が割り当てられるものとする。

```
int i = 10;
int * p;
p=&i;
// i の値と、値が置かれるアドレス値 を出力
printf("_____ \n", _____, _____);           // 出力      10      0x1000
// p の値（アドレス値） を出力
printf("_____ \n", _____);           // 出力      0x1000
```

12. いくつかのファイルがはいったディレクトリ dr をひとつのファイルにまとめるには、_____と言うコマンドを使って_____とする。こうして出来たファイルを、もとのディレクトリに戻すには_____とする。

13. 12の1ステップ目で作成したファイルを圧縮したい場合には、_____と言うコマンドを使って_____とする。こうして出来たファイルには、もとのファイル名に_____と言うサフィックスが付いている。圧縮されたファイルを元に戻すには_____と言うコマンドを使って_____とする。