問題1

構造体の宣言をせよ。構造体型名はCar、メンバはnum(整数),gas(実数)とする。

問題2

問題1で宣言した構造体型をつかった変数(構造体変数)car1を宣言せよ

問題3

問題1で、メンバnumに1234を代入し、gasに25.5を代入せよ。

問題4

問題2の構造体変数car1を宣言すると、メモリはどうなるか

問題5

問題2の構造体変数で、メンバnumに入れた整数を表示せよ。

問題6

問題2の構造体変数を次のように宣言できるようにせよ。

Car car1;

問題7

構造体変数を一気に初期化せよ。但しnumは1234、gasは25.5になるように。

問題8

問題1の構造体型の変数を考える。

car2にcar1をコピーせよ

問題9

問題2の構造体変数についてポインタを使う方法をいえ

答え1

struct Car

{

int num; /\*ナンバー\*/

double gas; /\*ガソリン残量\*/

};

注意：最後のセミコロンを忘れないこと。

答え2

struct Car car1;

答え3

car1.num=1234;

car1.gas=25.5;

(左辺の)「.」は「メンバ演算子」という

答え4

numとgasの組が確保される。

答え5

printf(“%d”,car1.num);

代入や、scanfなども、「(構造体変数名).(メンバ名)」を普通の変数のように扱うことをできる。

答え6

問題1を次のように書き換えればよい。

typedef struct Car

{

int num; /\*ナンバー\*/

double gas; /\*ガソリン残量\*/

} Car;

(main関数の外部でやるのがよい)

答え7

Car car1 = {1234, 25.5}

答え8

car2=car1;

各メンバを一つ一つコピーしなくてよい。

また、car2を初期化する場合にも同様なことができる。

Car car2 = car1;

答え9

Car\* pC

pC=&car1;

とすればポインタが扱える。

この時、car1.numと同じものは次の2通りで表せる。

(\*pC).num;

あるいは

pC->num;