REVIEW QUESTION

1.

int letter\_gradep(int points\*char gradep, \*char just\_missedp){

if(points >= 90 && points <= 100){

\*gradep = 'A';

}else if(points >= 80 && points <= 90){

\*gradep = 'B';

}else if(points >= 70 && points <= 80){

\*gradep = 'C';

}else if(points >= 60 && points <= 70){

\*gradep = 'D';

}else if(points >= 50 && points <= 60){

\*gradep = 'E';

}else if(points >= 40 && points <= 50){

\*gradep = 'F';

}else if(points >= 30 && points <= 40){

\*gradep = 'G';

}else if(points >= 20 && points <= 30){

\*gradep = 'H';

}else if(points >= 10 && points <= 20){

\*gradep = 'I';

}else if(points >= 0 && points <= 10){

\*gradep = 'J';

}else

if(points == 89){

\*just\_missedp = 'T';

}else f(points == 79){

\*just\_missedp = 'T';

}else f(points == 69){

\*just\_missedp = 'T';

}else f(points == 59){

\*just\_missedp = 'T';

}else f(points == 49){

\*just\_missedp = 'T';

}else f(points == 39){

\*just\_missedp = 'T';

}else f(points == 29){

\*just\_missedp = 'T';

}else f(points == 19){

\*just\_missedp = 'T';

}else f(points == 9){

\*just\_missedp = 'T';

}else{

\*just\_missedp = 'F';

}

return \*gradep, \*just\_missedp;

}

2. karena fungsi void tidak mengembailan nilai (?). alasan pribadi sebenarnya mungkin karena saya kurang mengerti tentang pemakaian fungsi void

3. Ketika fungsi dipanggil dan dieksekusi, komputer menyediakan suatu memori di data area fungsi untuk masing-masing parameter. Untuk input parameter, data area fungsi berisi nilai data yang sebenarnya, Untuk ouput parameter, hasil akan disimpan dalam pemanggilan fungsi.

4. fungsi happy

5.

Data area data area

Fungsi main fungsi trail

a a

3,98 3,98

b b

2,1 1,98

6. a. jika memungkinkan mengakses data yang berbeda, itu dapat manjadi fleksibel

b. parameter bisa digunakan sebagai masukan ke suatu fungsi sekaligus keluaran dari suatu fungsi, karena ini memang ada.