1. Какво представлява указателят ?

Указателят е специална променлива от даден тип, която за стойност има адрес на клетка от паметта, в която клетка може да се запише стойност от типа на указателя (ако указателят е от тип int, то в клетката може да се запише цяло число, ако указателят е от тип double, то в клетката може да се запише реално число, ако указателят е от тип char, то в клетката може да се запише символ).Например, ако имаме указател от тип int с име ptr (декларира се по следния начин: int\* ptr; ) и променлива от тип int (например int e = 7; ), то адресът на тази променлива (който се извлича (получава) чрез &e) може да бъде присвоен като стойност на указателя от тип int ptr (т.е. да се извърши присвояването ptr = &e; ). Накратко казано, указателят е специална променлива от даден тип, която приема за стойности адреси на клетки от паметта, в които клетки са записани променливи (или константи) от типа на указателя.

2. За указателите към константа и константните указатели:

2.1 Указател към константа -> не можем да променяме стойността намираща се в клетката, към която сочи указателят към константа. Самият указател към константа, обаче, можем да го пренасочим към друга клетка от паметта (т.е. можем да променим стойността на указателя, задавайки му адрес на друга клетка от паметта).

int e = 5;

const int\* a = &e;

(\*a)++; -> Груба грешка, тъй като a е указател към константа, а константи не могат да се променят !!!

a++; -> Това е позволено в случая, тъй като a НЕ е константен указател, а е указател към константа.

- Указателят към константа може да се декларира и по следния начин: int const\* a = &e; На семантично ниво няма разлика с другия тип декларация: const int\* a = &e;

2.2 Константен указател -> веднъж насочен, не можем да го пренасочваме, т.е. веднъж ако сме му задали адрес, то след това вече не можем да променяме този зададен му адрес, т.е. не можем да му сложим за стойност друг адрес.

int e = 5, r = 7;

int\* const a = &e;

a = &r; -> Груба грешка, тъй като a е константен указател, т.е. той е константа, а константите не могат да се променят, т.е. както виждаме в случая, след като сме присвоили на константния указател a адреса на променливата e като негова стойност, то след това не можем да променим стойността на този константен указател, както сме се опитали в случая, да му присвоим адреса на променливата r като нова негова стойност.

(пълна аналогия с обикновените константи от тип int, например ако сме имали int e = 5, r = 7; const int a = e; a = r; -> Груба грешка, тъй като a е константа от тип int и съответно веднъж инициализирана, не можем да и променяме стойността.)

Въпроси:

1. Можем ли да насочим указател от тип int към указател към константа от тип int ? -> не, дава грешка, тъй като заплашваме да променим съдържанието на клетката, към която е насочен указателят към константа.

int e = 5;

const int\* a = &e;

int\* b = a; //Тук ще даде грешка, тъй като за стойност на обикновения указател от тип int b слагаме (присвояваме) стойността на указател към константа от тип int a (тази стойност е адрес на клетка в паметта) и чрез обикновения указател от тип int b заплашваме да променим стойността, намираща се в самата клетка, чийто адрес е от своя страна стойност на указателя към константа от тип int a.

2. Можем ли да насочим указател от тип int към клетка от паметта, към която е насочен указател към константа от тип int ? -> да, можем, но това е лоша практика, тъй като с новонасочения указател от тип int можем да променим стойността на клетката, но към тази клетка е насочен и указател към константа.

int e = 5;

const int\* a = &e;

int\* b = &e; //Може, но е лоша практика, тъй като чрез указателя b можем да променим стойността намираща се в клетката, заделена за променливата e, а към нея сме насочили преди това указател към константа от тип int.