

# Programování v jazyce C pro chemiky (C2160) – závěrečné cvičení

## 10. Graf hydrofobicity a zabořenosti aminokyselin v proteinu

### Zadání

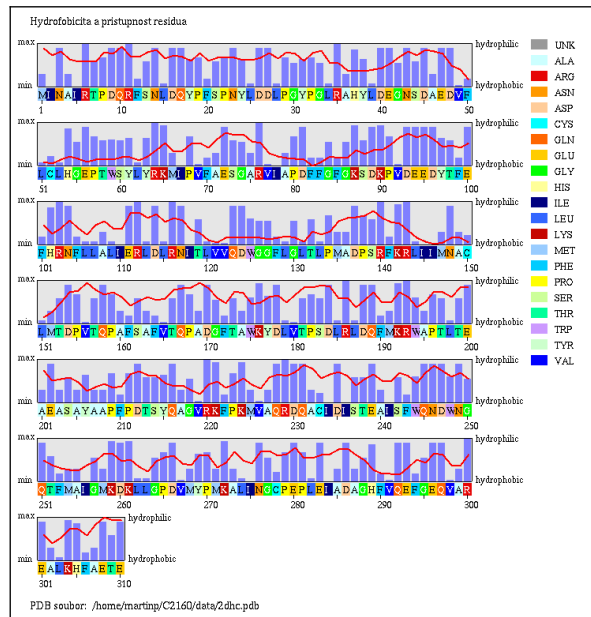
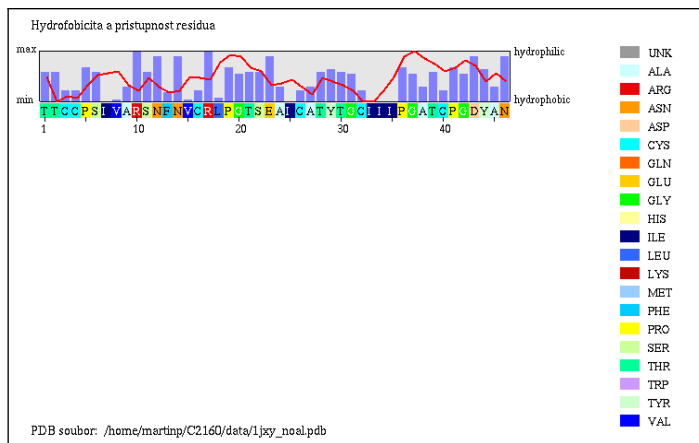
Vytvořte program, který zobrazí graf ukazující míru zabořenosti residua ve struktuře proteinu a zároveň hydrofobicitu postranních řetězců aminokyselin. Program bude mít následující vlastnosti:

- Bude načítat PDB soubor s proteinem (pouze standardní residua z řádků ATOM)
- Budou určeny C-alfa atomy aminokyselin
- Bude spočítat míru zabořenosti každého residua jako počet residuí nacházejících se do vzdálenosti 14 Å, vzdálenost residuí se aproximuje jako vzdálenost jejich C-alfa atomů
- Bude zobrazovat graf obsahující křivku charakterizující míru zabořenosti
- V grafu budou navíc ve formě sloupců zobrazeny hydrofobicity postranních řetězců aminokyselin
- Pod grafem bude barevně označen typ residua a uveden jeho písmenkový kód a čísla pořadí v sekvenci (viz. obrázek)
- Bude-li mít struktura více než 50 residuí, bude graf rozdělen na více grafů, každý s max. 50 residuí
- Graf bude obsahovat barevnou legendu, nadpis a jméno PDB souboru
- Jméno vstupního PDB souboru bude specifikováno jako parametr na příkazovém řádku
- Program bude uživatele informovat o chybě při otevření souboru, načítání konfiguračního souboru, překročení maximální přípustné velikosti polí a pod.
- Zdrojový kód programu bude opatřen komentáři

Nepovinné rozšíření (+5 bodů):

- Program bude načítat konfigurační soubor, ve kterém bude specifikováno jméno vstupního PDB souboru na řádku ve formátu "INPUT\_FILE = jmeno\_pdb\_souboru", dále bude v konfiguračním souboru na samostatném řádku specifikována velikost okna ve formátu "WINDOW\_SIZE = sirka, vyska"
- Název konfiguračního souboru bude předán programu jako parametr na příkazovém řádku

Program otestujte se strukturou crambinu (*1jxy\_noal.pdb*) a enzymu haloalkan dehalogenáza (*2dhc.pdb*), které najdete mezi studijními materiály v IS MU ve složce „data“.



### Dodržujte následující pravidla

- Dbejte na správné odsazování textu
- Pro reálné proměnné používejte typ `double`, ne `float`
- Při každém použití operátoru dělení si ujasněte, zdali dochází k celočíselnému nebo reálnému dělení a jaký typ dělení požadujete
- Proměnné vždy inicializujte vhodnou hodnotou
- Při použití funkcí pro práci s řetězci a při práci s poli dbejte na to, aby nedošlo k překročení velikosti pole
- Dobře zvažte, které proměnné budou lokální a které globální
- Názvy globálních proměnných volte tak, aby z nich byl jasný význam proměnné, volte raději delší názvy
- Názvy funkcí volte tak, aby z nich bylo jasné, jakou činnost funkce vykonává
- Pro překlad programů používejte nástroj `make` (tj. vytvořte si příslušný `Makefile`)
- Z programu odstraňte veškerý kód, který není nutný pro splnění zadání (např. pozůstatky z minulých cvičení, zakomentované části kódu). Ponechat můžete funkci pro zápis PDB
- Program nesmí při překladu vypisovat žádné varovné hlášky (při použití parametrů `-Wall -pedantic`)
- Na začátek programu umístěte stručný komentář obsahující jméno autora, rok vytvoření, popis funkce programu, parametry příkazového řádku, popř. formát konfiguračního souboru popisující činnost programu
- Všechny funkce a proměnné opatřete komentářem

## Návod

1. Upravte funkci pro načítání PDB souboru tak, že bude načítat pouze řádky ATOM a nikoliv HETATM.
2. Ve struktuře proteinu vyhledejte pro každé residuum atom C-alfa, tj. atom se jménem " CA " (vč. mezery na začátku a na konci).
3. Pro pohodlnější práci s C-alfa atomy přidejte do struktury RESIDUE proměnnou (např. `atom_c_alpha`) která bude obsahovat index atomu C-alfa v poli atomů (tj. pořadí v poli atomů). Hodnotu proměnné nastavte pro každé residuum ve funkci pro vyhledávání residuí nebo v samostatné funkci.
4. Pro pohodlnější práci se souřadnicemi C-alfa atomů přidejte do struktury RESIDUE proměnnou, která bude obsahovat souřadnice atomu C-alfa.
5. Vytvořte funkci (např. `get_points_distance()`), které předáte souřadnice dvou bodů a ona vrátí vzdálenost mezi těmito body. Tuto funkci využijete při výpočtu vzdálenosti mezi C-alfa atomy dvou residuí (případně může funkce přijímat jako argument indexy dvou residuí).
6. Do struktury RESIDUE přidejte celočíselnou proměnnou (pojmenovanou např. `accessibility`) která bude obsahovat hodnotu zanořenosti, tj. počet residuí nacházejících se ve vzdálenosti menší je 14 Å.
7. Spočítejte hodnotu zanořenosti tak, že pro každou dvojici residuí porovnáte vzájemnou vzdálenost jejich C-alfa atomů. Pokud je vzdálenost menší než 14 Å, zvětšete hodnotu proměnné o `accessibility` o 1 pro obě residua.
8. Hodnoty hydrofobicit (Kyte a Doolittle, 1982) najdete na: <https://web.expasy.org/protscale/pscale/Hphob.Doolittle.html>  
Hodnoty hydrofobicit přidejte do struktury obsahující barvy residuí a jejich hodnoty získávejte podobným mechanismem jako barvy residuí:

```
typedef struct
{
    char code3[4];           // three-letter code
    char code1;             // one-letter code
    double color_r;
    double color_g;
    double color_b;
    double hydrophobicity;
} RESIDUE_TYPE;

RESIDUE_TYPE residue_types[RESID_TYPES_COUNT] =
{
    {"UNK", 'X', 153/255.0, 153/255.0,
     153/255.0, 0.0},
    {"ALA", 'A', 204/255.0, 255/255.0,
     255/255.0, 1.8},
    {"ARG", 'R', 230/255.0, 6/255.0,
     6/255.0, -4.5},
    {"ASN", 'N', 255/255.0, 153/255.0,
     0/255.0, -3.5},
    ....
}
```

## Testovací data

Hodnoty zanořenosti residuí pro strukturu crambinu (`1jxy_noal.pdb`):

THR1: 26	ARG17: 28	ILE33: 40
THR2: 40	LEU18: 18	ILE34: 34
CYS3: 37	PRO19: 14	ILE35: 26
CYS4: 38	GLY20: 15	PRO36: 15
PRO5: 31	THR21: 21	GLY37: 12
SER6: 25	SER22: 23	ALA38: 16
ILE7: 24	GLU23: 31	THR39: 19
VAL8: 23	ALA24: 30	CYS40: 23
ALA9: 31	ILE25: 28	PRO41: 21
ARG10: 34	CYS26: 32	GLY42: 17
SER11: 27	ALA27: 36	ASP43: 20
ASN12: 32	THR28: 27	TYR44: 29
PHE13: 35	TYR29: 29	ALA45: 24
ASN14: 34	THR30: 31	ASN46: 29
VAL15: 26	GLY31: 34	
CYS16: 27	CYS32: 40	