# Některé typy úloh, které můžete očekávat na testu 3

# Příklad 1

1. Ve funkci main definujte dvě pole znaků o délce N (hodnotu N nastavte pomocí příkazu preprocesoru #define na hodnotu 255). Pole pojmenujte pole1 a pole2.
2. Napište funkci nacti, která ze standardního vstupu čte znaky a ukládá je postupně do pole. Po zadání znaku konce řádku (‘\n’) je čtení ukončeno. Funkce má jako dva parametry: ukazatel na pole a maximální délka pole. Návratovou hodnotou funkce je počet skutečně přečtených znaků. Při čtení kontrolujte, aby nebylo přečteno více znaků, než je velikost pole. V případě dosažení velikosti pole vypište na standardní výstup “Chyba – dosazeno max. velikosti vstupniho pole!“ a funkci ukončete.
3. Napište funkci tiskni, která vytiskne pole znaků o zadané délce.
4. Napište funkci obrat, která uloží obsah vstupního pole do výstupního pole, ale v opačném pořadí (ABC bude uloženo jako CBA). Vstupními parametry budou: ukazatel na vstupní pole, ukazatel na výstupní pole a skutečný počet znaků ve vstupním poli.
5. Úplné funkční prototypy funkcí nacti, tiskni, obrat umístěte do hlavičkového souboru vektor.h. Hlavičkový soubor ošetřete proti vícenásobnému vlažení. Soubor vektro.h použijte ve funkci main.
6. Ve funkci main zavolejte funkce v následujícím pořadí nacti, tiskni, obrat, tiskni.

# Příklad 2

1. Ve funkci definujte pole pojmenované data prvků typu int o velikosti N. Hodnotu N nastavte pomocí #define na 30. Pole inicializujte v definici na následující hodnoty:  
   3, 1, 5, 5, 4, 3, 2, 1, 0, 2, 3, 5, 3, 2, 5, 1, 2, 4, 3, 1, 4, 4, 2, 0, 1, 1, 0, 1, 3, 2.  
   Dále definujte pole celých čísel bez znaménka pojmenované cetnost a pole r\_cetnost prvků typu double. Obě pole mají velikosti 6 prvků.
2. Napište funkci vypCetnost, která bude mít tři parametry: ukazatel na pole se vstupními daty, ukazatel na pole, do kterého se uloží vypočtené četnosti a počet prvků vstupního pole. Funkce nebude nic vracet (vrací void). Funkce vypočte absolutní četnost (počet výskytů) čísel 0, 1, 2, 3, 4, 5 ve vstupním poli a výsledky uloží do výstupního pole. (Např. když se číslo 3 v poli data vyskytlo 6 krát, tak prvek cetnost[3] bude obsahovat hodnotu 6.)
3. Napište funkci vypRCetnost, která přepočítá hodnoty absolutní četnosti výskytu čísel 0 až 5 na relativní četnosti. Relativní četnost získáte podělením četnosti celkovým počtem prvků. (Např. relativní četnost výskytu čísla 3 je 6/30 = 0,2.) Funkce bude mít tři parametry: ukazatel na pole s absolutními četnostmi, ukazatel na pole s relativními četnostmi, počet prvků polí s četnostmi (zde je to 6).
4. Napište funkci tisk, která vytiskne absolutní a relativní četnosti výskytu prvků 0, 1, 2, 3, 4, 5. Relativní četnost tiskněte s přesností na 2 desetinná místa. Tištěný řádek bude vypadat následovně:  
   Cislo: 3 Absolutni cetnost: 6 Relativni cetnost: 0.20
5. Úplné funkční prototypy funkcí vypCetnost, vypRCetnost a tisk umístěte do hlavičkového souboru mujstat.h. Hlavičkový soubor ošetřete proti vícenásobnému vlažení. Soubor mujstat.h použijte ve funkci main.
6. Funkce vypCetnost, vypRCetnost a tisk zavolejte z funkce main.

# Příklad 3

Modifikujte příklad 2 tak, že vstupní data přečtete z příkazové řádky. Např. příkazová řádka bude obsahovat řetězec: 3155432102353251243144. Zde si musíte sami určit, kolik znaků číslic řetězec skutečně obsahuje (předpokládejte, že jich může být maximálně 30).

# Příklad 4

1. Ve funkci main definujte proměnné x, y, z reprezentují souřadnice bodu v prostoru.
2. Napište funkci cti, která přečte souřadnice bodu ze standardního vstupu a uloží je do proměnných x, y, z.
3. Definujte makro DELKA(X,Y,Z), které vypočítá délku vektoru.
4. Ve funkci main zavolejte funkci cti. Vytiskněte délku vektoru. Pro výpočet délky použijte makro DELKA.

# Příklad 5

1. Ve funkci main jsou dána 3 jednorozměrná pole o velikosti int poleA[M], int poleB[M] a int poleA[2\*M]. Hodnotu M nastavte pomocí #define na 8.
2. Napište funkci cti, která vyzve uživatele k zadání počtu prvků pole a ze standardního vstupu tento počet přečte (číslo typu int). Zkontrolujte, zda zadané číslo není větší než velikost pole. V případě, že ano, funkci ukončete s návratovou hodnotu -1. Jinak přečte zadaný počet prvků, prvky ukládejte do pole. Funkci bude předán ukazatel na pole a velikost pole. Funkce vrací počet skutečně přečtených prvků.
3. Napište funkci copy, která překopíruje prvky jednoho pole do pole druhého. Funkce bude mít 3 parametry: ukazatel na zdrojové pole, ukazatel na cílové pole a počet prvků, které se mají překopírovat. (Počet kopírovaných prvků může být menší než M.) Návratovou hodnotou funkce bude počet překopírovaných prvků.
4. Napište funkci, která vytiskne zadaný počet prvků pole. Funkce bude mít jako parametry ukazatel na pole a počet tištěných prvků.
5. Úplné funkční prototypy funkcí cti, copy a tisk umístěte do hlavičkového souboru pole.h. Hlavičkový soubor ošetřete proti vícenásobnému vložení.
6. Ve funkci main načtěte pomocí funkce cti hodnoty do poleA a do poleB (funkci volejte dvakrát). Pomocí funkce copy překopírujte uložený počet hodnot do z poleA do poleC. Následně za poslední uloženou hodnotu do poleC překopírujte obsah poleB. (Byl-li obsah poleA 1,2,3 a obsah poleB 4, 5, bude obsah poleC 1, 2, 3, 4, 5. Nakonec pomocí funkce tisk vytiskněte obsah poleC.

# Příklad 6

1. Ve funkci main definujte řetězce char strA[] = “abcd“ a char strB[] = “XYZ“ a pole char buf[20].
2. Napište následující funkce pro práci s řetězci:
   1. Funkci, která vrátí délku řetězce (počet znaků před ukončovací ‘\0’). Jako parametr je funkci předán ukazatel na začátek řetězce.
   2. Funkci, která překopíruje řetězec do pole znaků. Funkce bude mít jako parametry ukazatel na zdrojový řetězec a ukazatel na cílové pole. Funkce vrací ukazatel na zakončující ‘\0’ zkopírovaného řetězce.
   3. Funkci, která spojí dva řetězce do třetího. Parametry funkce: ukazatel na první řetězec, ukazatel na druhý řetězec, ukazatel na pole znaků, kam se má uložit spojený řetězec. Funkce vrací ukazatel na první znak spojeného řetězce.
   4. Funkci, která zjistí polohu prvního výskytu daného znaku v řetězci. Parametry funkce budou ukazatel na řetězec a hledaný znak. Návratovou hodnotou bude pozice hledaného znaku (index v poli).
   5. Funkci, která zjistí počet výskytů daného znaku v řetězci. Parametry funkce budou ukazatel na řetězec a hledaný znak. Návratovou hodnotou bude počet výskytů zadaného znaku.
   6. Funkci, která zjistí polohu prvního výskytu daného (pod)řetězce v řetězci. Parametry funkce budou ukazatel na řetězec a ukazatel na hledaný (pod)řetězec. Návratovou hodnotou bude pozice hledaného (pod)řetězce (index v poli).
   7. Funkci, která zjistí počet výskytů daného (pod)řetězce v řetězci. Parametry funkce budou ukazatel na řetězec a hledaný ukazatel na hledaný (pod)řetězec. Návratovou hodnotou bude počet výskytů zadaného (pod)řetězce.
3. Úplné funkční prototypy výše uvedených funkcí umístěte do hlavičkového souboru retezce.h. Hlavičkový soubor ošetřete proti vícenásobnému vložení.
4. Ve funkci main zavolejte výše uvedené funkce a ověřte jejich funkčnost.