

Zadání testu:**Jmeno:**

Napište funkci, která bude spojovat řetězce. Vícenásobným voláním této funkce vytvořte z několika řetězců jeden výsledný.

Obecná pravidla:

- testy provádějte v uvedeném pořadí
- manipulaci s pamětí provádějte až po provedení všech testů pro vstupní parametry
- chyby jsou signalizovány písmenem (V=hodnota, M=paměť) a číslem cyklu, ve kterém k chybě došlo.
- chyby hlásí checker (paměť a soubor – při nevhodné manipulaci ihned, jinak dává sumář na konci programu)
- pokud program spadne, jedná se většinou o špatnou manipulaci s pamětí (je nutné určit cyklus, při kterém k chybě došlo a vyzkoušet jaké parametry funkce dostala a jak na ně reagovala – trasování)

Návod:

- 1) Své funkce pište v souboru funkce3a.c s příslušným hlavičkovým souborem funkce3a.h. Hlavičkový soubor ošetřete. Volání svých funkcí provádějte z funkce Testování (tu překopírujte ze souboru s main). Po odladění svou funkci zaregistrujte způsobem, který je ukázán v souboru s main (registrujete funkci a druhý parametr je konstanta nadefinovaná v souboru test.h, která má ve svém názvu název registrované funkce).
- 2) Napište funkci AddText, která k řetězci, který dostane jako parametr (aVstup), přidá řetězec, který získá voláním funkce AllocGetText() (viz test.h). Pro sloučený řetězec vytvořte ve funkci novou paměť, kam sloučíte řetězce. Výsledek předejte pomocí místa na které ukazuje druhý z ukazatelů (aVystup).
Prototyp funkce je int AddText(char *aVstup, char **aVystup)

Funkce char *AllocGetText() naalokuje paměť pro text, naplní ji textem a vrátí ukazatel na tuto paměť jako návratovou hodnotu (zároveň předává i vlastnictví této paměti volající funkci). Funkci AllocGetText volejte až po provedení kontrol (které neodhalí chybový stav). Funkci AllocGetText pouze jednou na volání funkce AddText -

Není-li vstupní řetězec zadán a není-li ani platný text z funkce AllocGetText, vrátí se prázdný řetězec (řetězec obsahující pouze ukončovací znak).

Není-li vstupní řetězec zadán (aVstup má hodnotu NULL), bude výsledek roven řetězci z funkce AllocGetText.

Nevrátí-li AllocGetText platný text (vrátí ukazatel NULL), bude výsledkem vstupní řetězec.

Návratové hodnoty funkce AddText:

nezáporné číslo - délka načteného řetězce (bez ukončovacího znaku)

(testy na chyby provádějte v níže uvedeném pořadí)

- 1 (kterýkoli) předaný ukazatel nemá platnou adresu (tj. obsahuje hodnotu NULL)
- 2 na adrese odkazované ukazatelem pro uložení výsledku jsou již data (je tam platná adresa)
- 3 chyba při alokaci paměti (nedostatek paměti)

- 3) Napište funkci LongText (bez parametrů), která zavolá funkci AddText 12krát (poprvé s prázdným řetězcem). Vrácený řetězec použijte k opětovnému volání (tj. bude se stále připojovat k minulému výsledku). Po ukončení cyklu předejte výsledek (zavolejte s výsledným řetězcem) funkci ResText - řetězec tímto předáte funkci, která se stane jeho vlastníkem (odalokuje ho). Zajistěte, aby byla tato funkce napsána korektně.
Prototyp funkce void ResText(char *txt).
Pokud nastane chyba v této funkci, nebo ve funkci volané, odalokujte paměť a funkci ukončete.

Pozn.: Funkci AddText můžete při psaní funkce podle bodu 3 (nikoli 2) nahradit předdefinovanou funkcí fpzAddText