Курсовая работа

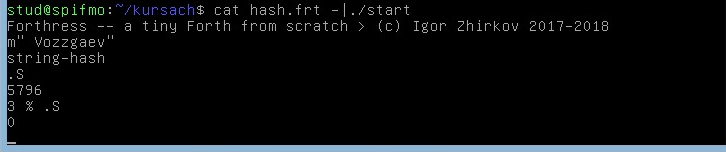
Предмет: СПО

I Этап

Автор:

Возжаев Артем Валерьевич

2019

Генерация варианта:

1) Программа, проверяющая число на четность

: parityCheck ( x - true/false )

2 % not if ." parity" else ." no parity" then

;

2) Программа, проверяющая на простоту

: primaryChechk ( x | if |x % i\_2..x == 0| => false else true )

dup 0 < if ." x < 0 !!" else ( x | x x | x dup - to save num)

dup 2 < if ." 1 - is prime" else

1 >r ( Pop from data stack into return stack - prepare to for )

repeat

dup ( save x )

1 r> + dup >r ( increment i in for and save it to x % i )

% 0 = ( chech end of for i=x)

until

r> = if ." Num is prime" else ." Non prime" then

then

then

;

3)Выделения памяти (allot) запись в нее значения и вывод адреса :

: simpleAllot ( x )

primaryChechk ( a - 1/0 ) 1 allot ( 1/0 - 1/0 adres ) dup rot swap ( 1/0 adres - adres 1/0 adres )

! ( adres 1/0 adres - adres ! - write 1/0 ro adres)

." -" ( print result and adress to store )

.

;

4)Слово, принимающее указатели на две строки и сливающие их

: concatstr ( srtadr1 stradr2 - srtadr3 )

( 1 need count length to each string - use count )

( to save adr duplicate var )

dup count ( srtadr1 stradr2 - srtadr1 stradr2 lengthstr2 )

( now need count length to str1 - rot to save adr duplicate var and count )

rot ( srtadr1 stradr2 lengthstr2 - srtadr2 lengthstr2 stradr1 )

dup count ( srtadr2 lengthstr2 stradr1 - stradr2 lengthstr2 srtadr1 lengthstr1 )

( now we need heap with size lengthstr1 + lengthstr2 + 1 but lengthstr1 save to make offset )

rot swap ( stradr2 lengthstr2 srtadr1 lengthstr1 - stradr2 srtadr1 lengthstr2 lengthstr1 )

dup rot + 1 + ( stradr2 srtadr1 lengthstr2 lengthstr1 - stradr2 srtadr1 lengthstr1 lengthstr3 )

heap-alloc ( stradr2 srtadr1 lengthstr1 lengthstr3 - stradr2 srtadr1 lengthstr1 stradr3 )

( need copy str1 to adr3 string-copy stradr3 srtadr1 before clean heap-str1 to save memory and save srtadr3 to find adr to str2 )

rot dup rot dup rot ( stradr2 srtadr1 lengthstr1 stradr3 - stradr2 lengthstr1 srtadr1 stradr3 stradr3 srtadr1 )

string-copy ( stradr2 lengthstr1 srtadr1 stradr3 stradr3 srtadr1 - stradr2 lengthstr1 srtadr1 stradr3 )

swap heap-free ( clean heap-str1 - stradr2 lengthstr1 stradr3 )

( last we need that write str2 to stradr3 + lengthstr1 and clean str2 save stradr2; save srtadr3 to return )

dup rot + ( stradr2 lengthstr1 stradr3 - stradr2 stradr3 stradr3 + lengthstr1 )

rot dup rot swap ( stradr2 stradr3 stradr3 + lengthstr1 - stradr3 stradr2 stradr3 + lengthstr1 stradr2 )

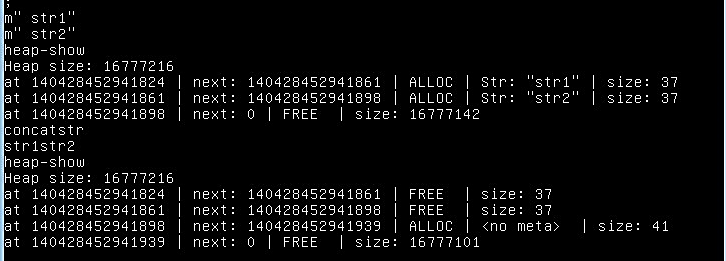
string-copy ( stradr3 stradr2 stradr3 + lengthstr1 stradr2 - stradr3 stradr2 )

heap-free ( clean heap-str2 => stradr3 )

prints

;

Проверка на утечку:



5)Вторая часть задания ( вариант 0 )

: kolatz ( x - xn xn-1 ... x1 x1 == 1 )

dup ( x - x x )

1 = ( x x - x 1/0 )

if . else

repeat ( while xn != 1 do )

dup dup ( x - x x x )

2 % ( x x x - x x 0/x%2 )

0 = ( x x 0/x%2 - x x 1/0 )

if ( x x )

2 / ( x x - x x/2 )

dup . cr

else

3 ( x x - x x 3 )

\* ( x x 3 - x x\*3 )

1 + ( x x\*3 - x x\*3 +1 )

dup . cr

then

dup 1 ( x x/2 - x x/2 x/2 1 | x x\*3 +1 x\*3 +1 1 )

= ( x x/2 1/0 | x x\*3 +1 1/0 )

until

then

;