**STRINGS**

 Report ZVAISH\_STRINGS.  
  
  
\*TRANSLATE STRINGS TO LOWER CASE OR UPPER CASE  
PARAMETERS P\_A(10).  
TRANSLATE P\_A TO LOWER CASE.  
write P\_A.  
  
 PARAMETERS P\_B(10) OBLIGATORY.  
 TRANSLATE P\_A TO UPPER CASE.  
 WRITE P\_B.  
  
 concatenate  
PARAMETERS : P\_A(10),P\_B(10).  
DATA: lv\_des(20) type c.  
CONCATENATE P\_A P\_B INTO lv\_des.  
write lv\_des.  
  
PARAMETERS: P\_A(10),P\_B(10),P\_C.  
DATA: lv\_des(20) type c.  
CONCATENATE P\_A P\_B INTO lv\_des SEPARATED BY P\_C.  
CONCATENATE P\_A P\_B INTO lv\_des SEPARATED BY SPACE. \* this is executed  
write lv\_des.  
  
SRRLEN  
  
PARAMETERS P\_A(10).  
DATA:lv\_des type c.  
lv\_des = STRLEN( P\_A ).  
write lv\_des.  
  
  
  
SPLIT  
PARAMETERS P\_A(10).  
DATA: lv\_des,lv\_des2.  
SPLIT P\_A at ',' INTO lv\_des lv\_des2.  
write / :lv\_des,lv\_des2.  
  
  
  
  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*New Strings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\* translate to uppercase  
  
PARAMETERS p\_A(12) type c.  
Translate P\_A to UPPER CASE.  
write p\_A.  
  
\*\*\*\*\*\*translate to lowercase\*\*\*\*\*  
PARAMETERS p\_B(12) type c.  
TRANSLATE p\_B to LOWER CASE.  
write p\_B.  
  
\*\*\*\*\*\*\*\*concatenate\*\*\*\*\*\*\*\*\*\*\*\*\*  
data lv\_c(20) type c.  
PARAMETERS p\_A(10) type c.  
PARAMETERS p\_b(10) type c.  
CONCATENATE p\_A p\_b into lv\_c .  
write lv\_c.  
  
\*\*\*\*\*\*\*concate separated by space  
  
data lv\_c(20) type c.  
PARAMETERS p\_a(10) type c.  
PARAMETERS p\_b(10) type c.  
concatenate p\_a p\_b into lv\_c SEPARATED BY space.  
write lv\_c.  
translate lv\_c to UPPER CASE.  
  
\*\*\*\*\*\*\*\*\*Concatenate by \*  
data lv\_c(20) type c.  
PARAMETERS p\_a(10) type c.  
PARAMETERS p\_b(10) type c.  
concatenate p\_a p\_b into lv\_c SEPARATED BY '\*'.  
write lv\_c.  
translate lv\_c to UPPER CASE.  
  
\*\*\*\*\*\*\*\*\*concatenate with user input  
  
data lv\_c(20) type c.  
PARAMETERS p\_a(10) type c.  
PARAMETERS p\_b(10) type c.  
PARAMETERS user type c.  
concatenate p\_a p\_b into lv\_c SEPARATED BY user.  
write lv\_c.  
translate lv\_c to UPPER CASE.  
  
  
\*\*\*\*\*\*\*\*\*\*\*split\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
PARAMETERS p\_split(10) type c.  
data : lv\_a(10) type c,  
      lv\_b(10) type c.  
split p\_split at '1' into lv\_a lv\_b.  
write /: lv\_a ,lv\_b.  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*offsetting(substring)\*\*\*\*\*\*\*\*\*  
  
PARAMETERS p\_a(20).  
  
write p\_a+0.  
data lv\_s(20).  
lv\_s(20) = p\_a+2(6).  
write lv\_s.  
  
\*\*\*\*\*\*\*\*strlen\*\*\*\*\*\*\*\*\*\*  
PARAMETERS p\_a(20) type c.  
data lv\_a(20) type c.  
lv\_a = strlen( p\_a ).  
write lv\_a.  
  
  
  
\*\*\*\*\*\*\*\*\*\*Replace\*\*\*\*\*  
PARAMETERS p\_a(20).  
data lv\_a(30).  
if p\_a CA 'I'.  
  REPLACE 'I' WITH 'O' INTO lv\_a. "into local variable not possibe only in p\_a  
    write lv\_a.  
  ENDIF.  
   write P\_a.  
  
   PARAMETERS p\_a(20).  
data lv\_a(20).  
if p\_a CA 'I'.  
  REPLACE 'I' WITH 'O' INTO p\_a. "into local variable directly is not possibe only in p\_a  
lv\_a = p\_a.  
  ENDIF.  
   write P\_a.  
   write lv\_a.