File svd88.s

```
#include "syscalnr.h"

.SECT .TEXT

!-------------------------!
!Inizio del programma!
!-------------------------!
start:
        MOV BP, SP                              !Inserisco sullo stack il base pointer

!------------------------------!
!Messaggio di Benvenuto!
!------------------------------!

        PUSH msgAst                             !Asterischi
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgIntro                           !Messaggio di introduzione
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgAst
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

!---------------------------------!
!Stampa del Menu Principale!
!---------------------------------!
menu:
        CALL initMenu

!-----------------------------------------------!
!Scelta dell'opzione dal Menu Principale!
!-----------------------------------------------!
scelta:
        PUSH _GETCHAR
        SYS
        ADD SP, 2

        MOVB BL, AL

        PUSH _GETCHAR
        SYS
        ADD SP, 2

        CMPB AL, 10
        JNE erroreScelta

        MOVB AL, BL

        CMPB AL, '1'
```

```
        JE statoCARICA
        CMPB AL, '2'
        JE filtra
        CMPB AL, '3'
        JE statoSALVA
        CMPB AL, '4'
        JE autoProg
        CMPB AL, '5'
        JE uscita

        JMP erroreScelta

!------------------------------------------------------------------!
!Caricamento del file input.txt nella variabile fotogramma!
!------------------------------------------------------------------!
statoCARICA:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgInitCarica
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH fotogramma
        CALL CARICA

        ADD SP, 2

        MOV DX, 0                                   !Rendo rifiltrabile il fotogramma

        PUSH msgOK
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH ritornaMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH _GETCHAR
        SYS
        ADD SP, 2

        JMP menu                                    !Ristampo il Menu Principale

!------------------------------------------------------------------------!
!Inizializzazione della lettura dei pixel dal vettore fotogramma!
!con conseguente inserimento sullo stack, creando cosi' la       !
!finestra di osservazione.                                       !
!------------------------------------------------------------------------!
filtra:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        MOV SI, 0
        MOV AX, fotogramma(SI)
```

```
        CMPB AL, 0                      !Se fotogramma non e' stato ancora caricato, lo rimando
al Menu con un messaggio
        JE noFiltra

        CMP DX, -1                      !Se fotogramma e' gia' stato filtrato, lo rimando al Menu
con un messaggio
        JNE initF

noFiltra:
        PUSH msgNoCaricato
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH ritornaMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH _GETCHAR
        SYS

        JMP menu

!-------------------------------------------------!
!Filtrare fotogramma estraendo i Mediani!
!-------------------------------------------------!
initF:
        PUSH msgInitFiltra
        PUSH _PRINTF
        SYS
        ADD SP, 4

        MOV DX, 0                       !Registro per la moltiplicazione nell'algoritmo
        MOV BX, 0                       !Registro per l'inserimento dei singoli pixel sullo stack
        MOV DI, 0
        DEC CX

letturaF:
        MOV SI, CX                      !Parto dall'ultimo carattere di fotogramma, leggendo
verso sinistra
        MOV AX, fotogramma(SI)          !Leggo in AX, carattere per carattere, il contenuto di
fotogramma

        DEC CX

        CMPB AL, 10                     !Quando trovo uno spazio, passo a leggere il valore
successivo
        JE proxFP

        CMP CX, -2                      !Se arrivo al primo carattere di fotogramma, ossia l'ultimo
da leggere,
        JE fineLetFot                   !Lo stampo sullo stack

        SUBB AL, 48

        MOVB AH, 0

        CMP DX, 0                       !Se e' la prima cifra del numero, la salvo in BL
        JNE sommaF                      !Altrimenti eseguo l'algoritmo di moltiplicazione

        INC DX
        MOVB BL, AL
```

```
        JMP proxF

!-------------------------------------------!
!Algoritmo di moltiplicazione              !
!Es. 352 = ((2) + (5 * 10) + (3 * 100))!
!-------------------------------------------!
sommaF:
        PUSH CX                             !Salvo il contatore dei caratteri sullo stack
        PUSH DX                             !Salvo il contatore delle cifre
        MOV CX, 0                           !E lo azzero poiche' una volta arrivato a -1 si creano dei
problemi come contatore
        MOV CX, DX
        MOV DX, 10
        JMP mulnum

!-----------------------------------------------------------!
!Ciclo per le cifre delle decine, centinaia, migliaia...!
!-----------------------------------------------------------!
mulnum:
        MUL DX
        MOV DX, 10
        LOOP mulnum

        POP DX
        POP CX                              !Ripristino il contatore
        ADD BX, AX
        INC DX
        JMP proxF

!-----------------------------------------------------------!
!Inserimento sullo stack del pixel letto da fotogramma!
!-----------------------------------------------------------!
proxFP:
        PUSH BX                             !E inserisco il pixel sullo stack
        MOV BX, 0                           !Pulisco nuovamente BX da ogni valore
        MOV DX, 0

        CMP DI, 8
        JE avanti
        INC DI                              !Contatore degli elementi sullo stack

avanti:
        JMP azzeraBX

!-------------------------------------------------------------!
!Se invece devo ancora finire di leggere il pixel, continuo!
!-------------------------------------------------------------!
proxF:
        MOV AX, 0
        CMP CX, -1                          !Se ho finito di scorrere fotogramma, esco
        JE proxFP
        JMP letturaF

!-----------------------------------------------------------------------!
!Una volta inseriti i primi 8 pixel letti da fotogramma sullo stack!
!procedo a calcolare il mediano                                        !
!-----------------------------------------------------------------------!
azzeraBX:
        MOV SI, 0

        PUSH DI                             !Inserisco il numero dei pixel presenti sullo stack in DIM
        PUSH numFormInt
        PUSH DIM
```

```
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        MOV DX, CX                      !Salvo il contatore dei caratteri

        PUSH msgFdOP
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH (DIM)                      !Passo DIM come valore a STAMPA
        CALL STAMPA                     !Stampo la finestra di osservazione prima del calcolo del
mediano

        POP (DIM)

        MOV CX, DX                      !Ripristino CX

        MOV BX, 0                       !Uso BX come contatore dei mediani
        MOV SI, 0
        MOV DX, (buf_mediani)

!-------------------------!
!Calcolo del Mediano!
!-------------------------!
        PUSH DI                         !Carico DIM con 8
        PUSH numFormInt
        PUSH DIM
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        PUSH DI                         !Salvo i due contatori sullo stack
        PUSH CX

        PUSH (DIM)
        CALL MEDIANO                    !Calcolo il mediano degli 8 valori passati per parametro

        POP (DIM)

        CMP CX, 7                       !Controllo per l'elemento shiftato: se si hanno 8 pixel
sullo stack, rimuovo l'ultimo
        JNE normalCX
        POP CX
        POP DI
        POP BP
        JMP dopoCX

normalCX:
        POP CX                          !Ripristino il contatore caratteri
        POP DI

dopoCX:
        PUSH CX                         !Risalvo CX sullo stack

        MOV SI, 0
        CMP mediani(SI), 0              !Se e' il primo mediano calcolato, lo inserisco
direttamente nella variabile mediani
        JNE nextPix

        PUSH AX                         !Inserisco il primo mediano all'inizio della variabile
mediani
```

```
        PUSH numFormInt
        PUSH mediani
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        PUSH acapo                      !Poi inserisco lo \n nella variabile di supporto
        PUSH strForm
        PUSH buf_mediani
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        PUSH DI

        MOV AX, 0
        MOV CX, -1
        MOV DI, mediani

        REPNZ SCASB                     !Scandisco la variabile mediani fino a trovare il valore "0"

        MOV SI, buf_mediani             !Cio' che verra' copiato in mediani
        MOV DI, mediani
        MOV BX, CX
        ADD BX, 2
        SUB DI, BX                      !Faccio puntare DI allo "0" trovato da REPNZ SCASB

        REP MOVSB                       !E ci piazzo lo \n

        POP DI

        POP CX                          !Ripristino il contatore dei caratteri
        MOV DX, 0                       !Registro per la moltiplicazione nell'algoritmo
        MOV BX, 0                       !Registro per l'inserimento dei singoli pixel sullo stack

        JMP letturaF

!-----------------------------------------------------!
!Gestione dei mediani calcolati dopo il primo!
!-----------------------------------------------------!
nextPix:
        PUSH AX                         !Inserisco il mediano in buf_mediani
        PUSH numFormInt
        PUSH buf_mediani
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        PUSH DI

        MOV AX, 0
        MOV CX, -1
        MOV DI, mediani

        REPNZ SCASB                     !Scandisco mediani fino a trovare un valore nullo

        MOV SI, buf_mediani
        MOV DI, mediani
        MOV BX, CX
        ADD BX, 2
        SUB DI, BX

        REP MOVSB                       !E inserisco in tale posizione il contenuto di buf_mediani,
```

```
ossia il mediano calcolato

        PUSH acapo                      !Poi inserisco lo \n nella variabile di supporto
        PUSH strForm
        PUSH buf_mediani
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        MOV AX, 0
        MOV CX, -1
        MOV DI, mediani

        REPNZ SCASB                     !Scandisco la variabile mediani fino a trovare il valore "0"

        MOV SI, buf_mediani             !Cio' che verra' copiato in mediani
        MOV DI, mediani
        MOV BX, CX
        ADD BX, 2
        SUB DI, BX                      !Faccio puntare DI allo "0" trovato da REPNZ SCASB

        REP MOVSB                       !E ci piazzo lo \n

        POP DI

        POP CX                          !Ripristino il contatore dei caratteri
        MOV DX, 0                       !Registro per la moltiplicazione nell'algoritmo
        MOV BX, 0                       !Registro per l'inserimento dei singoli pixel sullo stack

        CMP CX, -1
        JE fineLetFot                   !Se ho finito di leggere fotogramma, controllo gli ultimi
pixel

        JMP letturaF

!-------------------------------------------------------------------!
!Calcolo dei mediani degli ultimi pixel presenti sullo stack!
!-------------------------------------------------------------------!
fineLetFot:
        CMP DI, 1                       !Se ho finito di calcolare anche gli ultimi mediani, salvo
mediani in fotogramma
        JE mer2fot

        CMP DI, 8                       !Se sono presenti ancora 8 pixel sullo stack, ripeto il ciclo
del mediano normale (vedi sopra)
        JE jmpAzzBX

        MOV AX, CX                      !Salvo il contatore dei caratteri in AX
        MOV DX, DI                      !e quello dei pixel in DX

        MOV BX, DX                      !Preparo gli indici SI e DI per lo shift dei rimanenti pixel
sullo stack
        ADD BX, BX
        SUB BX, 2
        MOV SI, BX

        SUB BX, 2
        MOV DI, BX

        MOV BP, SP

        MOV BX, 0                       !Sistemo i contatori e gli indici per trovare gli elementi
sullo stack
```

```
        MOV CX, 0

shiftEnd:                                   !Shifto i numeri sullo stack verso il basso per far spazio al
nuovo pixel
        MOV BX, (BP)(DI)
        MOV (BP)(SI), BX
        SUB SI, 2
        SUB DI, 2
        INC CX

        CMP CX, DX
        JNE shiftEnd

        MOV CX, AX
        MOV DI, DX

        POP BP                              !Elimino l'elemento shiftato

!-------------------------------------------------------------------------------------------------------!
!Se sono presenti ancora 8 elementi sullo stack, decremento DI e ripeto il calcolo del mediano!
!-------------------------------------------------------------------------------------------------------!
jmpAzzBX:
        DEC DI
        JMP azzeraBX

!---------------------------------------------!
!Inserimento di mediani in fotogrammi!
!---------------------------------------------!
mer2fot:
        POP BP

        CALL fotogrammiUD                   !Aggiorno la variabile fotogrammi inserendovi il
contenuto di mediani

        ADD CX, 2                           !Incremento il contatore di 2 per non scrivere ulteriori
simboli
        MOV BX, CX                          !Converto il contatore dei caratteri di mediani da negativo
a positivo
        SUB CX, BX                          !-50 = 50 facendo -50 - 2 X (-50)
        SUB CX, BX                          !Mi ritrovo in CX il numero dei caratteri contati in
fotogramma per la SALVA

        MOV DX, -1                          !Registro per il controllo della doppia invocazione di
FILTRA dal Menu

        PUSH msgEndFiltra
        PUSH _PRINTF
        SYS
        ADD SP, 4

        MOV SI, 0
        MOV AX, varSM(SI)

        SUBB AL, 48

        CMPB AL, 4                          !Controllo se non e' stato azionato la quarta opzione del
Menu Principale
        JE autoSalva

        PUSH ritornaMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4
```

```
        PUSH _GETCHAR
        SYS
        ADD SP, 2

        JMP menu                                !Ritorno al Menu Principale

!-------------------------------------------------!
!Scrittura di fotogramma sul file output.txt!
!-------------------------------------------------!
statoSALVA:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgInitSalva
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH fotogramma                         !Passo fotogramma a SALVA
        CALL SALVA

        MOV CX, -1                              !Pulizia di mediani
        MOV SI, null
        MOV DI, mediani

        REP MOVSB

        MOV CX, 0                               !Azzera CX

        PUSH msgOK
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH ritornaMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH _GETCHAR
        SYS
        ADD SP, 2

        JMP menu

!---------------------------------------!
!Automatizza l'intero programma!
!---------------------------------------!
autoProg:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgAutomatizza
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgInitCarica
```

```
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH 4                          !Salvo l'opzione del Menu Automatizza per il controllo a
fine Filtra
        PUSH numFormInt
        PUSH varSM
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        PUSH fotogramma
        CALL CARICA                     !Carico fotogramma

        ADD SP, 2

        PUSH msgOK
        PUSH _PRINTF
        SYS
        ADD SP, 4

        JMP initF                       !Inizio il filtraggio

autoSalva:
        PUSH fotogramma
        CALL SALVA                      !Salvo fotogramma in output.txt

        ADD SP, 2

        PUSH msgOK
        PUSH _PRINTF
        SYS
        ADD SP, 4

!--------------------------!
!Uscita dal programma!
!--------------------------!
uscita:                                 !Uscita dal programma, ripristinando stack e registri
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgUscita
        PUSH _PRINTF
        SYS
        ADD SP, 4

        MOV SP, BP
        POP AX
        POP BX
        POP CX
        PUSH _EXIT
        SYS

!-------------------!
!File di sostegno!
!-------------------!
#include "statoCARICA.s"
#include "statoSALVA.s"
#include "errori.s"
#include "fzMediano.s"
```

```
#include "fotogrammaUpdate.s"
#include "bufferSTAMPA.s"
#include "Menu.s"

.SECT .DATA

null:
        .ASCIZ ""
input:
        .ASCIZ "input.txt"
output:
        .ASCIZ "output.txt"
erra:
        .ASCIZ "Errore durante l'apertura del file!\n"
errli:
        .ASCIZ "Errore durante la lettura del file input.txt!\n"
errlo:
        .ASCIZ "Errore durante la lettura del file output.txt!\n"
errc:
        .ASCIZ "Errore durante la creazione del file output.txt!\n"
errch:
        .ASCIZ "Errore durante la chiusura del file!\n"
errscelta:
        .ASCIZ "Attenzione! L'opzione scelta non e' presente nel Menu.\n"
msgAst:
        .ASCIZ "**********************************************************************\n"
msgIntro:
        .ASCIZ "* Benvenuto nel sistema di elaborazione di segnali video digitali SVD88 *\n"
msgAstMenu:
        .ASCIZ "**********\n"
msgMenu:
        .ASCIZ "* Menu Principale *\n"
msgCarica:
        .ASCIZ "* 1. Carica       *\n"
msgFiltra:
        .ASCIZ "* 2. Filtra       *\n"
msgSalva:
        .ASCIZ "* 3. Salva        *\n"
msgAuto:
        .ASCIZ "* 4. Automatizza  *\n"
msgEsci:
        .ASCIZ "* 5. Esci         *\n"
msgInsMenu:
        .ASCIZ "Inserire l'opzione da eseguire: "
msgInitCarica:
        .ASCIZ "Caricamento del file input.txt in corso... "
msgOK:
        .ASCIZ "OK\n"
msgNoCaricato:
        .ASCIZ "Attenzione! Il file input.txt non e' stato ancora caricato, oppure e' gia' stato filtrato.\n"
msgInitFiltra:
        .ASCIZ "Applicazione del filtro in corso... \n"
msgEndFiltra:
        .ASCIZ "Filtro applicato con successo!\n"
ritornaMenu:
        .ASCIZ "Premi il tasto INVIO per ritornare al Menu principale"
msgInitSalva:
        .ASCIZ "Salvataggio di fotogramma nel file output.txt in corso... "
msgAutomatizza:
        .ASCIZ "Procedura di automatizzazione iniziata!\n"
msgUscita:
        .ASCIZ "Uscita.\n"
msgFdOP:
```

```
        .ASCIZ "Stampa della Finestra di Osservazione prima del calcolo del mediano.\n"
msgFdOD:
        .ASCIZ "Stampa della Finestra di Osservazione dopo il calcolo del mediano.\n"
numFormInt:
        .ASCIZ "%d"                                     !Formato intero
acapo:
        .ASCIZ "\n"
numAcapo:
        .ASCIZ "%d\n"
strForm:
        .ASCIZ "%s"                                     !Formato stringa
fotogramma:
        .SPACE 28800

.SECT.BSS

DIM_FOTOGRAMMA:
        .SPACE 28800
buf_mediani:                                            !Variabile di supporto per l'inserimento dei mediani
        .SPACE 12
varSM:                                                  !Variabile di controllo per la scelta del Menu
        .SPACE 1
mediani:                                                !Variabile contenente temporaneamente i mediani
calcolati
        .SPACE 30000
DIM:
        .SPACE 1
inputfd:
        .SPACE 2
outputfd:
        .SPACE 2
pulBuf:
        .SPACE 1
buf_lettura:
        .SPACE 1
```

File statoCARICA.s

```
!-------------------------------------------------------------!
!Carica in fotogramma il contenuto del file input.txt!
!leggendo un carattere alla volta e salvando nel      !
!registro CX il numero dei caratteri letti.           !
!-------------------------------------------------------------!
CARICA:
        PUSH BP                                 !Metto il BP sullo stack per ripristinare il RET della
CALL
        MOV BP, SP                              !Salvo lo SP nel BP

        PUSH 0                                  !Apro il file input.txt in modalita' di sola lettura
        PUSH input
        PUSH _OPEN
        SYS
        ADD SP, 6

        CMP AX, -1                              !Controllo se ci sono errori nell'apertura del file input.txt
        JE erroreApertura                       !Se ci sono, avverto ed esco dal programma

        MOV (inputfd), AX                       !Aggiorno il file descriptor del file input.txt

        MOV AX, 0

        MOV DI, fotogramma                      !Salvo in DI il vettore fotogramma che verra' caricato di
un carattere alla volta

        MOV CX, -1                              !Uso CX per contare quanti caratteri leggo dal file
input.txt
                                                !Inizializzo CX a -2 per leggere tutti i caratteri del file
senza gli spazi finali

!-------------------------------------------------------------------------------------!
!Leggo l'intero file input.txt, salvando il contenuto nel vettore fotogramma.!
!Leggo carattere per carattere, fino a raggiungere la fine del file.         !
!-------------------------------------------------------------------------------------!
leggi:
        MOV SI, buf_lettura                     !Salvo in SI il buffer della lettura che occorrera' a leggere
il contenuto di input.txt

        PUSH 1                                  !Leggo un carattere alla volta
        PUSH buf_lettura
        PUSH (inputfd)
        PUSH _READ
        SYS
        ADD SP, 8

        CMP AX, -1                              !Se e' occorso un errore, ripeto la lettura con altri 2
tentativi
        JE erroreLI

        CMP AX, 0                               !Se ho letto tutto il file, esco dalla lettura
        JE fineLeggi

        PUSH AX                                 !Salvo il contenuto di AX sullo stack

        LODSB
        STOSB                                   !Carico il carattere letto dal file input.txt nel vettore
fotogramma

        POP AX                                  !Ripristino AX
```

```
        INC CX                          !Incremento CX, il contatore dei caratteri letti dal file
input.txt

        JMP leggi

fineLeggi:
        MOV AX, 0

        PUSH (inputfd)                  !Se tutto e' andato a buon fine, chiudo il file
        PUSH _CLOSE
        SYS
        ADD SP, 4

        CMP AX, -1
        JE erroreChiusura               !Se e' occorso un errore durante la chiusura del file, lo
segnalo ed esco

        MOV AX, 0

        POP BP
        RET
```

File statoSALVA.s

```
!-------------------------------------------------------------------------------!
!Salvataggio del contenuto di fotogramma all'interno del file output.txt!
!-------------------------------------------------------------------------------!
SALVA:
        PUSH BP                                 !Metto il BP sullo stack per ripristinare il RET della
CALL
        MOV BP, SP                              !Salvo lo SP nel BP

        PUSH 0777                               !Creo, o sovrascrivo se gia' presente, un nuovo file vuoto
output.txt
        PUSH output                             !Dando i privilegi per la lettura, scrittura e cancellazione
del file
        PUSH _CREAT
        SYS
        ADD SP, 6

        CMP AX, -1                              !Controllo se ci sono errori la creazione del file output.txt
        JE erroreCreazione                      !Se ci sono, avverto ed esco dal programma

        MOV (outputfd), AX                      !Aggiorno il file descriptor del file output.txt

        MOV AX, 0

        PUSH 1                                  !Apro il file output.txt in modalita' di sola scrittura
        PUSH output
        PUSH _OPEN
        SYS
        ADD SP, 6

        CMP AX, -1                              !Controllo se ci sono errori nell'apertura del file output.txt
        JE erroreApertura                       !Se ci sono, avverto ed esco dal programma

        MOV (outputfd), AX                      !Aggiorno il file descriptor del file input.txt

        MOV AX, 0

!-------------------------------------------------------------------------!
!Salvo il contenuto del vettore fotogramma nel file output.txt  !
!Se occorre un errore, faccio altri 2 tentativi                 !
!Se anche al terzo tentativo non funziona, esco dal programma!
!-------------------------------------------------------------------------!
scrivi:
        PUSH CX                                 !Scrivo sul file output.txt i CX caratteri contati durante la
lettura del file input.txt
        PUSH fotogramma
        PUSH (outputfd)
        PUSH _WRITE
        SYS
        ADD SP, 8

        CMP AX, -1                              !Se e' occorso un errore, ripeto la scrittura con altri 2
tentativi
        JE erroreLO

        MOV AX, 0

        PUSH (outputfd)                         !Se tutto e' andato a buon fine, chiudo il file
        PUSH _CLOSE
        SYS
        ADD SP, 4
```

```
        CMP AX, -1
        JE erroreChiusura                   !Se e' occorso un errore durante la chiusura del file, lo segnalo ed esco

        MOV AX, 0

        POP BP
        RET
```

File fzMediano.s

```
!-------------------------------------------------------------------------!
!Calcolo del mediano all'interno della finestra d'osservazione!
!-------------------------------------------------------------------------!
MEDIANO:
        PUSH BP
        MOV BP, SP
        ADD BP, 4                       !Faccio in modo che BP punti a DIM, sullo stack

        MOV AX, (BP)                    !Prendo DIM sullo stack e lo metto in AX

        SUBB AL, 48                     !Lo trasformo in numero intero e lo sposto in AH
        MOVB AH, AL

        ADD BP, 6                       !Torno al primo elemento sullo stack

        MOVB AL, 0                      !Sono gli indici per i cicli
        MOVB BL, 0

        MOVB BH, AH
        DECB BH                         !DIM-1
        MOV CX, 0
        MOV DX, 2

min1:
        MOV SI, 0                       !Sono gli indici per trovare i pixel sullo stack
        MOV DI, 2
        INCB BL                         !Ciclo esterno
        CMPB BL, BH                     !Se BL e' maggiore di DIM-1 allora termina il ciclo
        JG med

        MOVB AL, BL

min2:   INCB AL                         !Ciclo interno
        CMPB AL, AH
        JG min1                         !Se AL e' maggiore di DIM torna al ciclo esterno

        MOV DX, (BP)(SI)
        CMP DX, (BP)(DI)                !Se l'elemento sullo stack e' maggiore del successivo
allora passo al prossimo elemento
        JG incr                         !incrementando prima gli indici di BP

        MOV DX, (BP)(SI)                !Se no, scambio la posizione dei due elementi
        MOV CX, (BP)(DI)
        MOV (BP)(SI), CX
        MOV (BP)(DI), DX

incr:   ADD SI, 2                       !Incremento i contatori
        ADD DI, 2
        JMP min2                        !Ripeto il ciclo interno

med:
        MOV SI, 0                       !Calcolo il mediano
        MOV AX, DIM(SI)
        SUB AX, 48                      !Metto in AX il contenuto di DIM
        MOVB CH, 2
        DIVB CH
        CMPB AH, 0
        JNE meddispari

        MOV BX, 2                       !Se DIM e' pari recupero l'elemento alla posizione
```

```
DIM/2+1
        MUL BX
        SUB AX, 2
        MOV SI, AX
        MOV AX, (BP)(SI)
        JMP 1f

meddispari:
        MOV BX, 2                       !Se DIM e' dispari recupero l'elemento alla posizione
DIM/2
        MOVB AH, 0
        MUL BX
        MOV SI, AX
        MOV AX, (BP)(SI)


1:      MOV SI, 12

        PUSH AX                         !Salvo il valore del mediano sullo stack

        PUSH msgFdOD                    !Stampo a video il messaggio della stampa dopo il calcolo
del mediano
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH (DIM)
        CALL STAMPA                     !Invoco la stampa dopo aver ordinato gli elementi

        POP (DIM)

        POP AX                          !Recupero il valore del mediano sullo stack
        MOV BP, SP

        CMPB CH, 8                      !Se non ci sono 8 elementi sullo stack, non shiftare
        JNE esci

        MOV BX, 0                       !Sistemo i contatori e gli indici per trovare gli elementi
sullo stack
        MOV CX, 0
        MOV SI, 24
        MOV DI, 22

shift:                                  !Shifto i numeri sullo stack verso il basso per far spazio al
nuovo pixel
        MOV BX, (BP)(DI)
        MOV (BP)(SI), BX
        SUB SI, 2
        SUB DI, 2
        INC CX

        CMP CX, 7
        JNE shift

esci:
        POP BP
        RET
```

File bufferSTAMPA.s

```
!-------------------------------------------------------------------!
!Tale funzione permette di stampare a video lo stato della!
!finestra d'osservazione (lo stack) quando viene chiamata!
!-------------------------------------------------------------------!
STAMPA:
        PUSH BP
        MOV BP, SP

        ADD BP, 4
        MOV CX, (BP)

        SUBB CL, 48
        MOVB CH, CL

        ADD BP, 2

        MOVB CL, 1

ciclo:
        CMPB CL, CH
        JG esciciclo
        MOV BX, (BP)(SI)
        PUSH BX
        PUSH numAcapo
        PUSH _PRINTF
        SYS
        ADD SP, 6
        ADD SI, 2
        INCB CL
        JMP ciclo


esciciclo:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        POP BP
        RET
```

File fotogrammaUpdate.s

```
!----------------------------------------------!
!Inserimento di mediani in fotogramma!
!----------------------------------------------!
fotogrammiUD:
        PUSH BP
        MOV BP, SP

        MOV BX, 0

        MOV SI, 0
        MOV AX, varSM(SI)

        SUBB AL, 48

        CMPB AL, 4                      !Controllo se non e' stato azionata la quarta opzione del
Menu Principale
        JNE aggFot

        MOVB BL, AL                     !Salvo in BX il valore

aggFot:
        MOV AX, 0
        MOV CX, -1
        MOV DI, mediani

        REPNZ SCASB                     !Conto di quanti caratteri e' formato mediani, ossia quanti
pixel mediani contiene

        MOV SI, mediani
        MOV DI, fotogramma

        PUSH mediani                    !Metto il contenuto di mediani in fotogramma
        PUSH strForm
        PUSH fotogramma
        PUSH _SPRINTF
        SYS
        ADD SP, 8

        CMPB BL, 4                      !Se l'Automatizza e' stata invocata, riaggiorno la variabile
varSM
        JNE fineAgg

        PUSH 4                          !Salvo l'opzione del Menu Automatizza per il controllo a
fine Filtra
        PUSH numFormInt
        PUSH varSM
        PUSH _SPRINTF
        SYS
        ADD SP, 8

fineAgg:
        POP BP
        RET
```

File errori.s

```
!-------------------------------------------------!
!Gestione dell'errore dell'apertura dei file!
!-------------------------------------------------!
erroreApertura:
        MOV AX, 1

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH erra                               !Avverto dell'errore
        PUSH _PRINTF
        SYS
        ADD SP, 4

        JMP uscita

!-----------------------------------------------------------!
!Gestione dell'errore della lettura del file input.txt!
!-----------------------------------------------------------!
erroreLI:
        MOV AX, 1

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH errli                              !Avverto dell'errore
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH (inputfd)                          !Chiudo il file input.txt
        PUSH _CLOSE
        SYS
        ADD SP, 4

        JMP uscita

!-------------------------------------------------!
!Gestione dell'errore della creazione del file!
!-------------------------------------------------!
erroreCreazione:
        MOV AX, 1

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH errc                               !Avverto dell'errore
        PUSH _PRINTF
        SYS
        ADD SP, 4

        JMP uscita
```

```
!-------------------------------------------------------------!
!Gestione dell'errore della lettura del file output.txt!
!-------------------------------------------------------------!
erroreLO:
        MOV AX, 1

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH errlo                              !Avverto dell'errore
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH (outputfd)                         !Chiudo il file input.txt
        PUSH _CLOSE
        SYS
        ADD SP, 4

        JMP uscita

!-------------------------------------------------------------!
!Gestione dell'errore durante la chiusura del file!
!-------------------------------------------------------------!
erroreChiusura:
        MOV AX, 1

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH errch                              !Avverto dell'errore
        PUSH _PRINTF
        SYS
        ADD SP, 4

        JMP uscita

!-----------------------------------------------------------------------!
!Gestione dell'errore durante la scelta dell'opzione del Menu!
!-----------------------------------------------------------------------!
erroreScelta:
        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH errscelta
        PUSH _PRINTF
        SYS
        ADD SP, 4

        JMP uscita
```

File Menu.s

```
!-----------------------------------------------------!
!Stampa del Menu visualizzabile dall'utente!
!-----------------------------------------------------!
initMenu:
        PUSH BP
        MOV BP, SP

        PUSH msgAstMenu                 !Asterischi Menu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgMenu                    !Titolo Menu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgAstMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgCarica                  !Funzione CARICA
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgFiltra                  !Funzione MEDIANO
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgSalva                   !Funzione SALVA
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgAuto                    !Automatizza l'intero processo del progetto
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgEsci                    !Uscita dal programma
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgAstMenu
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH acapo
        PUSH _PRINTF
        SYS
        ADD SP, 4

        PUSH msgInsMenu                 !Richiesta d'inserimento dell'opzione da eseguire
        PUSH _PRINTF
```

```
SYS
ADD SP, 4

POP BP
RET
```