```
PROGRAM StudentManagement

ENUMERATION
    SUCCESS = 0
    FAILED = 1

STRUCTURE Students
    CHAR nomor_registrasi[11]
    CHAR nama[70]
    INT umur

FUNCTION clrscr
    EXECUTE clear screen command
END FUNCTION

FUNCTION addStudent
    DECLARE Students newStudent
    DECLARE CHAR inputlagi
    DECLARE SIZE_T counter = 0
    OPEN file "db.csv" for appending
    IF file is open THEN
        DO
            ALLOCATE memory for newStudent
            IF memory allocation fails THEN
                PRINT error message
                SET status to FAILED
            ENDIF

            CALL clrscr
            PRINT prompt for registration number
            READ input into newStudent[counter].nomor_registrasi

            PRINT prompt for student name
            READ input into newStudent[counter].nama

            PRINT prompt for student age
            READ input into newStudent[counter].umur

            PRINT prompt for adding more students
            READ input into inputlagi
            INCREMENT counter
        WHILE inputlagi is 'y' or 'Y'

        FOR i from 0 to counter-1 DO
            WRITE newStudent[i] to file
        END FOR

        FREE memory allocated for newStudent
        CLOSE file
    ELSE
        PRINT error message
        SET status to FAILED
    ENDIF
END FUNCTION
```

```
FUNCTION displayStudent
    DECLARE Students savedStudent
    DECLARE SIZE_T counter = 0
    DECLARE CHAR bufferStudents[256]
    OPEN file "db.csv" for reading
    IF file is open THEN
        SET records to countRecords(file)
        ALLOCATE memory for savedStudent

        IF memory allocation fails THEN
            PRINT error message
            SET status to FAILED
        ENDIF

        REWIND file
        PRINT header for student data
        WHILE READ line from file into bufferStudents DO
            PARSE bufferStudents into savedStudent[counter]
            INCREMENT counter
        END WHILE

        PRINT total records
        FOR i from 0 to counter-1 DO
            PRINT savedStudent[i] details
        END FOR

        FREE memory allocated for savedStudent
        PRINT prompt to return to main menu
        CLOSE file
    ELSE
        PRINT error message
        SET status to FAILED
    ENDIF
END FUNCTION

FUNCTION searchStudent
    DECLARE CHAR input_nomor_registrasi[12]
    DECLARE CHAR buffer[256]
    DECLARE SIZE_T counter = 0
    DECLARE Students savedStudent
    OPEN file "db.csv" for reading
    IF file is open THEN
        PRINT prompt for registration number
        READ input_nomor_registrasi

        SET records to countRecords(file)
        ALLOCATE memory for savedStudent

        IF memory allocation fails THEN
            PRINT error message
            SET status to FAILED
        ENDIF
        REWIND file
        WHILE READ line from file into buffer DO
            PARSE buffer into savedStudent[counter]
            INCREMENT counter
        END WHILE

        FOR i from 0 to records-1 DO
            IF input_nomor_registrasi matches
savedStudent[i].nomor_registrasi THEN
                PRINT savedStudent[i] details
            ENDIF
        END FOR

        FREE memory allocated for savedStudent
        CLOSE file
        PRINT prompt to return to main menu
    ELSE
        PRINT error message
        SET status to FAILED
    ENDIF
END FUNCTION

FUNCTION deleteStudent
    DECLARE CHAR input_nomor_registrasi[11]
    DECLARE CHAR buffer[256]
    DECLARE SIZE_T counter = 0
    DECLARE SIZE_T found = 0
    DECLARE Students savedStudent
    DECLARE Students notDeletedStudent
    OPEN file "db.csv" for reading and writing
    IF file is open THEN
        PRINT prompt for registration number
        READ input_nomor_registrasi

        SET records to countRecords(file)
        ALLOCATE memory for savedStudent and notDeletedStudent

        IF memory allocation fails THEN
            PRINT error message
            SET status to FAILED
        ENDIF

        REWIND file
        WHILE READ line from file into buffer DO
            PARSE buffer into savedStudent[counter]
            INCREMENT counter
        END WHILE

        CLOSE file
        DECLARE SIZE_T counternotdeleted = 0
        FOR i from 0 to records-1 DO
            IF input_nomor_registrasi does not match
savedStudent[i].nomor_registrasi THEN
                COPY savedStudent[i] to
```

```
notDeletedStudent[counternotdeleted]
                INCREMENT counternotdeleted
            ELSE
                PRINT savedStudent[i] details
                INCREMENT found
            ENDIF
        END FOR

        IF found == 0 THEN
            PRINT "Data tidak ditemukan"
        ELSE
            OPEN file "db.csv" for writing
            IF file is open THEN
                FOR i from 0 to counternotdeleted-1 DO
                    WRITE notDeletedStudent[i] to file
                END FOR
            ELSE
                PRINT error message
                SET status to FAILED
            ENDIF
        ENDIF

        FREE memory allocated for savedStudent and notDeletedStudent
        CLOSE file
        PRINT prompt to return to main menu
    ELSE
        PRINT error message
        SET status to FAILED
    ENDIF
END FUNCTION

FUNCTION countRecords(FILE *dbcount)
    DECLARE SIZE_T counter = 0
    DECLARE CHAR buffer[256]
    WHILE READ line from dbcount DO
        INCREMENT counter
    END WHILE
    RETURN counter
END FUNCTION

FUNCTION setMsgNotif(const char *msg)
    ALLOCATE memory for msg_notif
    COPY msg into msg_notif
END FUNCTION

MAIN FUNCTION
    DECLARE INT navmenu = 0
    WHILE navmenu != 5 DO
        CALL clrscr
        PRINT program header
        IF msg_notif is not NULL THEN
            PRINT msg_notif
            FREE msg_notif
        ENDIF
        PRINT menu options
        READ navmenu
        SWITCH navmenu DO
            CASE 1:
                CALL addStudent
            CASE 2:
                CALL displayStudent
            CASE 3:
                CALL searchStudent
            CASE 4:
                CALL deleteStudent
            DEFAULT:
                PRINT "Pilihan tidak valid."
        END SWITCH
    END WHILE
    RETURN status
END PROGRAM
```