Consider the data type TIME

Requirements

Declaration

Operations :

Relational Operators

Input Statements

Output Statements

Conditional Statements

Looping

Passing as parameter to function

Return

Requirement 2: Memory

Single variable: 2 byte for hour: 2 bytes for Min and 2 bytes for seconds Or Sizeof (int) for hour: sizeof (int) for Min: sizeof(int) Seconds

Multiple variable declaration

TIME t1[10];

Each location is of the form Sizeof (int) for hour: sizeof (int) for Min: sizeof(int) Seconds

Only stack and static memory is required. Heap is not required.

Requirement Specification - Declaration

Requirement 1: Syntax

Single variable declaration

TIME t1,t2;

Multiple variable declaration

TIME t1[10];

Requirement Specification - Initialization

• When Declared: Initialized to zero to each location • User Initialization: • T1= [10:12:34] • Or • T1= [10:12] • Or • T1=[10] • T1=[::36] • or • T1=[:13:56] • or • T1=[:12:]

Requirement Specification : Array

Declaration: Syntax is shown. Example: TIME T1[number]

Memory: number \*sizeof(hour)\*sizeof(minute)\*sizeof(seconds)

Initialization: Example: TIME T1[number].

• When Declared: Initialized to zero to each location • User Initialization: • T1[i]={ [10:12:34], [10:12],[10],[::36],[:13:56],[:12:]}

T1[i]={ [10:12:34], [10:12],[10],[::36],[:13:56],[:12:]}

T1[0].hr T1[0].min T1[0].sec T1[1].hr T1[1].min T1[2].hr T1[3].sec T1[4].min T1[4].sec T1[5].sec

T1[0] T1[1]

T1[2] T1[3] T1[4]

T1[5]

6 bytes 4 bytes 2 bytes 2 bytes 2 bytes 4 bytes

Requirement Specification : Input and Output statements

Choice1: Read(T1) Write(T1)

User Perspective: 1: Simple 2: Readable 3: Writable User should be aware that 3 values need to be read.

Choice 2: Read(T1.hr, T1.min, T1.sec) Read(T1.hr) Read(T1.min) Read(T1.sec)

Write(T1.hr, T1.min, T1.sec) Write(T1.hr) Write(T1.min) Write(T1.sec)

Requirement 3: Relational Operator (==, <, >, <=, >=, !=)

Choice 1: T1== T2

Choice 2: T1.h==T2.h && T1.m==T2.m&&T1.s==T2.s

Choice 3:T1.h:T1.m:T1.s==T2.h:T2.m:T2.s

Choice 1 User Perspective

i: Satisfy Simplicity, Redability and Writability same as comparing two int variables.

Choice 1 Compiler Designer Perspective

Compiler should recognize individual members of TIME Allocate memory individually Comparison is not straight forward, individual members need to be compared as ISA has instructions such as JE, JLE, JNZ……etc. Designing compiler is a challenging part.

Requirement Specification : Operations

Limitations: Individual members cant be compared

Requirement 3: Relational Operator (==, <, >, <=, >=, !=)

Choice 1: T1== T2

Choice 2: T1.h==T2.h && T1.m==T2.m&&T1.s==T2.s

Choice 3:T1.h:T1.m:T1.s==T2.h:T2.m:T2.s

Choice 2 User Perspective

i: Not simple, Not Readable, Not Writable. Take more time to write the program.

Choice 2 Compiler Designer Perspective

It is easy for the Compiler to recognize individual members of TIME. Allocate memory individually Comparison is straight forward, identifying individual members to compare need no extra coding.

Requirement Specification : Operations

Requirement 3: Relational Operator (==, <, >, <=, >=, !=)

Choice 1: T1== T2

Choice 2: T1.h==T2.h && T1.m==T2.m&&T1.s==T2.s

Choice 3:T1.h:T1.m:T1.s==T2.h:T2.m:T2.s

Choice 3 User Perspective

i: Not simple, Not Readable, Not Writable. Take more time to write the program.

Choice 4 Compiler Designer Perspective

It is easy for the Compiler to recognize individual members of TIME. Allocate memory individually Comparison is straight forward, identifying individual members to compare need no extra coding.

Requirement Specification : Operations

Requirement Specification: Loop

Untill

If T1 is array Example: Untill i in T1

Countdown

If T1 is a single variable • countdown T1.min • countdown T1.sec • countdown T1.hr • countdown T1

If T1 is a single variable • i.min=0 Untill i.min in T1 • i.sec=0 Untill i.sec in T1 • i.hr=0 Untill i.hr in T1

Test Case 1

void program begin TIME T1; Read(T1); Write (T1); End

Test Case 2

void program begin TIME T1[10]; int I; i=0 untill T1 Read(T1); i=0 until T1 Write (T1); End

Test Case 3

void program begin TIME T1[10]; int I; i=0 untill T1 Read(T1);

countdown T1.min Write(T1);

End

Test Case 4

void function Compare begin parameter Time T1; Time T2 Read(T2); check(T1==T2) Write(“Equal”) otherwise Write(“Not Equal”)’ End

void program begin TIME T1; Read(T1); Call Compare(T1); End