**PART I**

|  |  |
| --- | --- |
| Line in file | Contents |
| 30 | int \* someIDs, theFirst, \*r; |
| 110 | someIDs =GetSomeIDs(); /\* defined below \*/ |
| 111 | theFirst = someIDs [0]; |
| 112 | r= ReorderIDs(someIDs); |
| 113-150 | /\* we want to use ‘theFirst’ and ‘r’ here\*/ |
|  |  |
| 499 | /\*-------- GetSomeIDs-----\*/ |
| 500 | int \* GetSomeIDs() |
| 501 | { |
| 502 | int ids[8]; |
| 503-550 | /\* The ids are defined here \*/ |
| 551 | return ids; |
| 552 | } |

1. Is GetSomeIDs() a reasonable function?

*1. It’s function contain error. Array ids defined as local and can be available only in procedure body. Inspite of function return address of this array but really it’s only pointer to memory area which can be rewrite in any time. Conclusion: this function can provide to access violation errors in program*

*2. This function doesn’t return count of items and doesn’t setup end pointer(‘\0’) for array.*

2. Is there a different way to write line 500 which preserves the same effective prototype? If so, what is it?

*I can rewrite this function with parameters which transfer by link.*

*void GetSomeIDs( int\* &val )*

*{*

*…*

*if ( !val )*

*val = new int[8];*

*…*

*}*

*example of run this function*

*int \*pVal = NULL;*

*GetSomeIDs( pVal );*

*After started this function pVal will contain new address(pointer to new memory vector). You should free memory in the end of use.*

3. What will ‘theFirst’ contain after line 111 is executed? Is this deterministic? Why?

*It will contain first value of array IDS, but if function GetSomeIDs wouldn’t rewrite it’s value can be absolutely random because array ids defined as local and can be available only in procedure body. Inspite of function return address of this array but really it’s only pointer to memory area which can be rewrite in any time.*

4. What line(s) not given should be provided for compilation?

*Line 112 because function ReorderIDs isn’t defined*

5. Is GetSomeIDs() good practice, and why?

*Why not. If you want defined array value in separate function you can do it, but don’t forget about right work with pointers.*

6. Might the code shown execute at all?

*If will be defined function ReorderIDs you can execute this code but result of it work will be unpredictable*

7. Correct the problems with GetSomeIDs(), and add some additional functionality to it, as follows. A single new version of the function should be provided.

a) Maintain the same "int \*" return type which returns a pointer to fixed sized array of ints.

b) IN ADDITION to its regular function return, provide to its calling functions a usable array of pointers to aliasID structures. The length of this array of pointers is returned by a call to GetNumberOfAliases(), which you may call only from within GetSomeIDs().

c) Use the structure and the functions defined below:

typedef struct {

char\* alias; /\* '\0'-terminated C string \*/

int specific\_id;

} aliasID;

/\* How many structures should be pointed to by the array

\*/

int GetNumberOfAliases(void);

/\* Get a pointer to the next structure. The structure itself

\* will be filled with data.

\* Caller is responsible for the cleanup of the returned structure

\* and its content. The latter are allocated in

\* dynamic memory.

\*/

aliasID \* **GetNextAlias**(void);

Do NOT(!) use "C++" syntax or language constructs.

This should be written in plain “C”.

Use good programming practice, as much as these instructions allow.

*Maybe I don’t understood questions but my mind code below can realize functionality described in your question*

*typedef struct {*

*char\* alias; /\* '\0'-terminated C string \*/*

*int specific\_id;*

*} aliasID;*

*int \*array = NULL;*

*/\* How many structures should be pointed to by the array*

*\*/*

*int GetNumberOfAliases(void)*

*{*

*if ( array ) {*

*int i = 0;*

*while ( array [i++] != 0 );*

*return i;*

*}*

*return -1;*

*}*

*//---------------------------------------------------------------------------*

*/\* Get a pointer to the next structure. The structure itself*

*\* will be filled with data.*

*\* Caller is responsible for the cleanup of the returned structure*

*\* and its content. The latter are allocated in*

*\* dynamic memory.*

*\*/*

*//---------------------------------------------------------------------------*

*aliasID \* GetNextAlias(void)*

*{*

*aliasID \*paliasID = (aliasID \*)malloc (sizeof(aliasID));*

*if ( array ) {*

*int vCnt = GetNumberOfAliases();*

*array = (int \*)realloc ( array, (vCnt + 1) \* sizeof(int) );*

*if (array) {*

*array[ vCnt - 1 ] = (int) paliasID;*

*array[ vCnt ] = 0;*

*} else {*

*free( (aliasID \*) paliasID );*

*return NULL;*

*}*

*}*

*else {*

*array = (int \*)malloc ( sizeof (int) \* 2 );*

*if ( array ) {*

*array[0] = (int) paliasID;*

*array[1] = 0;*

*} else {*

*free( (aliasID \*) paliasID );*

*return NULL;*

*}*

*}*

*return paliasID;*

*}*

*//---------------------------------------------------------------------------*

*int\* GetSomeIDs(int arrSize, int &vSize )*

*{*

*for (int i = 0; i < arrSize; i++ ) {*

*aliasID \* val = GetNextAlias();*

*if ( val ) {*

*val->alias = new char [10];*

*sprintf(val->alias, "val%d\0", i);*

*val->specific\_id = i;*

*} else*

*printf("Error memory allocate");*

*}*

*vSize = GetNumberOfAliases() - 1;*

*return array;*

*}void main ()*

*{*

*int vSize = 0;*

*int \*val = GetSomeIDs( 8, vSize );*

*if ( vSize == 8)*

*printf ("all is good");*

*}*

8 . Write a function which calls GetSomeIDs() as described in (7) above, prints out all data returned by it, and cleans up resources allocated in dynamic memory.

For both (7), and (8), above, DO NOT use "C++" syntax or language contructs. These should be written in plain “C”.

Use good programming practice, as much as these instructions allow.

*void FreeArray()*

*{*

*int vSize = 0, i = 0;*

*int \*val = GetSomeIDs( 2, vSize );*

*char specific\_id\_buf[11] = {'\0'};*

*while ( val[i] != 0 ) {*

*printf( ((aliasID \* )val[i])->alias );*

*free (((aliasID \* )val[i])->alias);*

*((aliasID \* )val[i])->alias = NULL;*

*memset(specific\_id\_buf, '\0', 11);*

*sprintf( specific\_id\_buf, "specific\_id = %d", ((aliasID \* )val[i])->specific\_id );*

*aliasID \*pVal = (aliasID \* )val[i];*

*free( pVal );*

*val[i++] = 0;*

*}*

*free ((int \*) val);*

*val = NULL;*

*}*

9. If you know SQL, please, as simply as possible, design a single table to hold orders made by customers. You may assume details (including names) about customers are held in some other table. Using the table that you designed for orders, and this other table, present a single query that gives the number of orders for every customer, one line per customer.

create table TORI.Clients

(

clientID int identity (1,1),

cName varchar(100) not null,

cSurname varchar(100) not null,

cBirthDate datetime not null

)

create index idx\_ClientID on TORI.Clients (clientID)

create table TORI.Goods

(

goodID int identity(1,1),

gName varchar(255) not null,

gAmount int not null

)

create index idx\_GoodsID on TORI.Goods (goodID)

create table TORI.Orders

(

orderID int identity(1,1),

oDate varchar(255) not null,

clientID int not null,

orderBodyID int not null,

orderStateID int not null

)

create index idx\_OrderID on TORI.Orders (orderID)

create table TORI.OrdersBody

(

orderBodyID int identity (1,1),

orderID int not null,

goodsID int not null,

gPrice numeric(22,4) not null

)

create index idx\_OrderID\_Body on TORI.OrdersBody (orderID)

select t2.clientID,

max(t2.cName + ' ' + t2.cSurname) as Client,

count(t1.orderID) as ordersCount

from TORI.Orders t1

inner join TORI.Clients t2

on t1.clientID = t2.clientID

group by t2.clientID

10. If you know Python, please answer the following:

There's a problem with the

following Python 2.x code, please fix it.

print reduce(lambda x, y: x+y, filter(lambda x: x%2, map(lambda x: x\*x, xrange

(10\*\*6)))) = sum(x\*x for x in xrange(1, 10\*\*6, 2))

After the fix, what would be printed? Explain the result.

*I never work with Python, but I can find decision if I spend some more time*

A. What resources (references texts or URLs) did you use to help you answer these questions and what help did they provide?

B. What help did others provide towards answering these questions?

**PART II**

1. Write a “C” function that modifies each element of a user supplied array by applying a user-provided function to each element of an array of integers. The user-provided function can also accept parameters. Please provide a typedef for the user-provided function.

*typedef void (\*anyfunc)( int &val );*

*void UserFunc ( int &val ){ val++; }*

*void RecalcArray(int \*val, anyfunc func )*

*{*

*int i = 0;*

*while ( val[i] != 0 ) {*

*func (val[i++]);*

*}*

*}*

*void main()*

*{*

*int val[10] = {1,2,3,4,6,7,8,9,0};*

*RecalcArray (val, UserFunc);*

*}*

2. Write a “C” function to convert an array to a doubly-linked list.

*ListID\* AddToList ( ListID \*pList, int arr\_val )*

*{*

*if ( pList ) {*

*ListID \*cList = new ListID;*

*cList->arr\_id = arr\_val;*

*cList->prevVal = pList;*

*cList->nextVal = NULL;*

*pList->nextVal = cList;*

*return cList;*

*} else {*

*pList = new ListID;*

*pList->arr\_id = arr\_val;*

*pList->prevVal = NULL;*

*pList->nextVal = NULL;*

*return pList;*

*}*

*}*

*ListID\* ConvertArrayToList( int \*array )*

*{*

*int i = 0;*

*ListID\* lHead = NULL;*

*while ( array[i] != 0 )*

*lHead = AddToList ( lHead, array[i++] );*

*return lHead;*

*}*

*void main ()*

*{*

*int val[10] = {1,2,3,4,5,6,7,8,9,0};*

*ListID \*lPointer = ConvertArrayToList( val ); // return pointer on last list item*

*}*

3. What resources (references texts or URLs) did you use to help you answer these questions and what help did they provide?

4. What help did others provide towards answering these questions?