```
%mem1 = getelementptr inbounds %struct.jpeg_common_struct,
                 ... %struct.jpeg_common_struct* %cinfo, i64 0, i32 1
                  %0 = bitcast %struct.jpeg_memory_mgr** %mem1 to %struct.my_memory_mgr**
                  %1 = load %struct.my_memory_mgr*, %struct.my_memory_mgr** %0, align 8, !tbaa
                 ... !3
                  %cmp = icmp ugt i64 %sizeofobject, 999999976
                  br i1 %cmp, label %if.then, label %if.end, !prof !9
                                                                                 F
                                  if.end:
                                   %rem = and i64 %sizeofobject, 7
                                   %cmp2 = icmp eq i64 %rem, 0
                                   %sub = sub nsw i64 8, %rem
                                   %add = select i1 %cmp2, i64 0, i64 %sub
                                   %sizeofobject.addr.0 = add i64 %add, %sizeofobject
                                   %5 = icmp ugt i32 %pool_id, 1
                                   br i1 %5, label %if.then7, label %if.end11, !prof!16
                                                                         F
                                             T
if.then7:
%err = getelementptr inbounds %struct.jpeg_common_struct,
... %struct.jpeg_common_struct* %cinfo, i64 0, i32 0
%6 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
...!tbaa!10
%msg code = getelementptr inbounds %struct.jpeg error mgr,
... %struct.jpeg_error_mgr* %6, i64 0, i32 5
store i32 12, i32* %msg_code, align 8, !tbaa !11
%arrayidx = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %6, i64 0, i32 6, i32 0, i64 0
store i32 %pool_id, i32* %arrayidx, align 4, !tbaa !14
%7 = bitcast %struct.jpeg_error_mgr* %6 to void
... (%struct.jpeg_common_struct*)**
%8 = load void (%struct.jpeg_common_struct*)*, void
... (%struct.jpeg_common_struct*)** %7, align 8, !tbaa !15
tail call void %8(%struct.jpeg_common_struct* nonnull %cinfo) #5
br label %if.end11
                       if.end11:
                       %add12 = add i64 %sizeofobject.addr.0, 24
                       %call = tail call i8* @jpeg_get_large(%struct.jpeg_common_struct* nonnull
                       ... %cinfo, i64 %add12) #5
                       %cmp13 = icmp eq i8* %call, null
                       br i1 %cmp13, label %if.then14, label %if.end15, !prof !9
                      if.end15:
                       %total_space_allocated = getelementptr inbounds %struct.my_memory_mgr,
                       ... %struct.my_memory_mgr* %1, i64 0, i32 5
                       %12 = load i64, i64* %total_space_allocated, align 8, !tbaa !17
                       %add17 = add i64 %12, %add12
                       store i64 %add17, i64* %total_space_allocated, align 8, !tbaa !17
                       %idxprom = sext i32 %pool_id to i64
                       %arrayidx18 = getelementptr inbounds %struct.my_memory_mgr,
                       ... %struct.my_memory_mgr* %1, i64 0, i32 2, i64 %idxprom
                       %13 = bitcast %union.large_pool_struct** %arrayidx18 to i64*
                       %14 = \text{load } i64, i64* \%13, align 8, !tbaa !20
                       %15 = bitcast i8* %call to i64*
                       store i64 %14, i64* %15, align 8, !tbaa !21
                       %bytes_used = getelementptr inbounds i8, i8* %call, i64 8
                       %16 = bitcast i8* %bytes_used to i64*
                       store i64 %sizeofobject.addr.0, i64* %16, align 8, !tbaa !23
                       %bytes_left = getelementptr inbounds i8, i8* %call, i64 16
                       %17 = bitcast i8* %bytes_left to i64*
                       store i64 0, i64* %17, align 8, !tbaa !24
                       %18 = bitcast %union.large_pool_struct** %arrayidx18 to i8**
                       store i8* %call, i8** %18, align 8, !tbaa !20
                       %add.ptr = getelementptr inbounds i8, i8* %call, i64 24
```

entry:

ret i8\* %add.ptr