```
entry.upper:
  %0 = load i32, i32* @_ZL9heartbeat, align 4, !tbaa !5
  %inc = add i32 %0, 1
  store i32 %inc, i32* @_ZL9heartbeat, align 4, !tbaa !5
  %conv = uitofp i32 %0 to float
  %arrayidx = getelementptr inbounds float, float* %ckpt_mem, i64 1
  store float %conv, float* %arrayidx, align 4, !tbaa !9
  %conv1 = sitofp i32 %id to float
  %arrayidx2 = getelementptr inbounds float, float* %ckpt_mem, i64 2
  store float %conv1, float* %arrayidx2, align 4, !tbaa !9
  %conv3 = sitofp i32 %i to float
  %arrayidx4 = getelementptr inbounds float, float* %ckpt_mem, i64 3
  store float %conv3, float* %arrayidx4, align 4, !tbaa !9
  %mul = mul i32 %size, %size
  br label %checkpoint.restoreControllerBB
```

```
checkpoint.restoreControllerBB:
  %load.CheckpointID_address = alloca i8, align 1
  %loaded.CheckpointID = load i8, i8* %load.CheckpointID_address, align 1
  switch i8 %loaded.CheckpointID, label %entry.lower [
  ]
```
def

```
entry.lower:
  %cmp17.not = icmp eq i32 %mul, 0
  br i1 %cmp17.not, label %for.cond.cleanup, label %for.body.preheader
```
T   F

```
for.body.preheader:
  %wide.trip.count = zext i32 %mul to i64
  br label %for.body.preheader.saveBB.id0
```

```
for.body.preheader.saveBB.id0:
  %store.ckpt_mem_address = alloca float*, align 8
  store float* %ckpt_mem, float** %store.ckpt_mem_address, align 8
  %store.result_address = alloca float*, align 8
  store float* %result, float** %store.result_address, align 8
  %store.wide.trip.count_address = alloca i64, align 8
  store i64 %wide.trip.count, i64* %store.wide.trip.count_address, align 8
  br label %for.body.preheader.junctionBB.id0
```

```
for.body.preheader.restoreBB.id0:
  %load.ckpt_mem_address = alloca float*, align 8
  %loaded.ckpt_mem = load float*, float** %load.ckpt_mem_address, align 8
  %load.result_address = alloca float*, align 8
  %loaded.result = load float*, float** %load.result_address, align 8
  %load.wide.trip.count_address = alloca i64, align 8
  %loaded.wide.trip.count = load i64, i64* %load.wide.trip.count_address,
  ... align 8
  br label %for.body.preheader.junctionBB.id0
```

```
for.body.preheader.junctionBB.id0:
  %new.ckpt_mem = phi float* [ %ckpt_mem, %for.body.preheader.saveBB.id0 ], [
  ... %loaded.ckpt_mem, %for.body.preheader.restoreBB.id0 ]
  %new.result = phi float* [ %result, %for.body.preheader.saveBB.id0 ], [
  ... %loaded.result, %for.body.preheader.restoreBB.id0 ]
  %new.wide.trip.count = phi i64 [ %wide.trip.count,
  ... %for.body.preheader.saveBB.id0 ], [ %loaded.wide.trip.count,
  ... %for.body.preheader.restoreBB.id0 ]
  br label %for.body.upper
```

```
for.body.upper:
  %indvars.iv = phi i64 [ 0, %for.body.preheader.junctionBB.id0 ], [
  ... %indvars.iv.next, %for.body.lower ]
  %arrayidx5 = getelementptr inbounds float, float* %result, i64 %indvars.iv
  %1 = load float, float* %arrayidx5, align 4, !tbaa !9
  %2 = add nuw nsw i64 %indvars.iv, 4
  %arrayidx7 = getelementptr inbounds float, float* %ckpt_mem, i64 %2
  store float %1, float* %arrayidx7, align 4, !tbaa !9
  %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1
  br label %for.body.lower
```

```
for.body.lower:
  %exitcond.not = icmp eq i64 %indvars.iv.next, %wide.trip.count
  br i1 %exitcond.not, label %for.cond.cleanup, label %for.body.upper,
  ... !llvm.loop !11
```
T   F

```
for.cond.cleanup:
  ret void
```

CFG for 'checkpoint' function