```
int main(int argc, char * argv[]) {
 int NIMGS;
 char **images;

 …

 images = (char **) malloc (sizeof(char *)*NIMGS);
 for (int i=0; i<NIMGS; i++) {
  images[i] = (char *) malloc (sizeof(char)*20);
  sprintf(images[i],"images/image%d.png", i);

void * res1 = readImage(image[i]);

void * res2 = processImage(res);

 …

}
```

```
class CPU_Stage: public ff_node {
 …

class ff_first_stage: public ff_node {
 …
I

int main(int argc, char * argv[]) {
 int NIMGS;
 char **images;

 ….

 images = (char **) malloc (sizeof(char *)*NIMGS);
 for (int i=0; i<NIMGS; i++) {
  images[i] = (char *) malloc (sizeof(char)*20);
  sprintf(images[i],"images/image%d.png", i);
 }

 StreamGen streamgen(NIMGS,images);

 ff_pipeline pipe;
 pipe.add_stage(&streamgen);
 pipe.add_stage(new ff_pipe_first_stage);
 pipe.add_stage(new CPU_Stage);

 …

}
```

```
class CPU_Stage: public ff_node {
 …
class ff_first_stage: public ff_node {
 …

int main(int argc, char * argv[]) {
 int NIMGS;
 char **images;

 …

 images = (char **) malloc (sizeof(char *)*NIMGS);
 for (int i=0; i<NIMGS; i++) {
  images[i] = (char *) malloc (sizeof(char)*20);
  sprintf(images[i],"images/image%d.png", i);
 }

 ff_farm<> global_farm;
 global_farm.add_collector(NULL);
 std::vector<ff_node*> gw;
 for (int i=0; i<nworkers; i++)
  gw.push_back(new ff_pipe_first_stage);
 global_farm.add_workers(gw);

 StreamGen streamgen(NIMGS,images);

 ff_pipeline pipe;
 pipe.add_stage(&streamgen);
 pipe.add_stage(&global_farm);
 pipe.add_stage(new CPU_Stage);

 …

}
```