```
fonction  getNearbyFunSearchByInputOnly (Date date, ArrayList names, String address, String
favourateFood) : return ArrayList

String[] out = getCoordinate(address);
address = getAddress(out);

firstfunSearchResults = getNearbySearch(out[0], out[1], average, type1, “");
firstfunSearchResult = getNearestPlace(address,firstfunSearchResults);
funSearchResults.add(firstfunSearchResult);
out = getCoordinate(firstfunSearchResult.getAddress());
address = firstfunSearchResult.getAddress();

secondfunSearchResults = getNearbySearch(out[0], out[1], average, type2, favourateFood);
secondfirstfunSearchResult = getNearestPlace(address,secondfunSearchResults);
funSearchResults.add(secondfirstfunSearchResult);
out = getCoordinate(secondfirstfunSearchResult.getAddress());
address = secondfirstfunSearchResult.getAddress();

thirdfunSearchResults = getNearbySearch(out[0], out[1], average, type3, "");
FunSearchResult thirdfunSearchResult = getNearestPlace(address,thirdfunSearchResults);
funSearchResults.add(thirdfunSearchResult);

return funSearchResults;



fonction getCoordinate(String addr) : return String []


                URL url = null;
                String html_output = null;
                url = new URL("https://maps.google.com/maps/api/geocode/json?address="+addr
+"&sensor=false&key=");

                        while (scan.hasNext()){
                                html_output += scan.nextLine();
                        scan.close();
                }

                JSONObject j = new JSONObject(html_output);
                System.out.println (j.toString());

                JSONArray o2 = (JSONArray) j.get("results");
                JSONObject o3 =  (JSONObject) o2.get(0);
                JSONObject o4 = (JSONObject) o3.get("geometry");
                JSONObject o5 = (JSONObject)o4.get("location");
                latLng[0]=  o5.get("lat").toString();
                latLng[1]=  o5.get("lng").toString();

                return latLng;


fonction getAddress(String[] latLng) : return String

                        url = new URL("https://maps.google.com/maps/api/geocode/json?
latlng="+latLng[0]+","+latLng[1]+"&sensor=false&key=");
```

```
                        while (scan.hasNext())
                                html_output += scan.nextLine();
                        scan.close();
                }

                JSONObject j = new JSONObject(html_output);
                print  (j.toString());
                JSONArray o1 = (JSONArray)j.get("results");

                JSONObject o2 = (JSONObject)o1.get(0);
                address = String.valueOf(o2.get("formatted_address"));

                return address;
        }



fonction  getNearbySearch(String lat, String lng, int m, String type, String title) : return ArrayList


                        url = new URL("https://maps.googleapis.com/maps/api/place/nearbysearch/
json?location="+lat+","+lng+"&radius="+m+"&type="+type+"&name="+title+"&key=" +
GooglePlacesKey);

                        while (scan.hasNext())
                                html_output += scan.nextLine();
                        scan.close();
                }

                JSONObject j =  JSONObject(html_output);
                print(j.toString());
                JSONArray result = j.getJSONArray("results");

                pour (int i = 0 ; i < result.length() ; i++){
                        JSONObject lieu = result.getJSONObject (i);
                        print ("  -> " + lieu.getString ("name") + ", " + lieu.getString ("vicinity"));
                        String name = lieu.getString("name");
                        String address = lieu.getString("vicinity");
                        funSearchResults.add(new FunSearchResult(name, address));
                }
                return funSearchResults;




fonction  calculateTimeCost(String address_a,String address_b) : return int

                        address_a = java.net.URLEncoder.encode(address_a, "UTF-8");
                        address_b = java.net.URLEncoder.encode(address_b, "UTF-8");
                        url = new URL("https://maps.googleapis.com/maps/api/distancematrix/
json?"+"origins="+address_a+"&destinations="+address_b+"&key=" + GooglePlacesKey);
                while (scan.hasNext()){
                                html_output += scan.nextLine();
                        scan.close();
                }
                JSONObject j = new JSONObject(html_output);
                print(j.toString());

                JSONArray o2 = (JSONArray) j.get("rows");
                JSONObject o3 =  (JSONObject) o2.get(0);
                JSONArray o4 = (JSONArray) o3.get("elements");
                JSONObject o5 =  (JSONObject) o4.get(0);
                JSONObject o6 =  (JSONObject) o5.get("duration");

                cost = o6.get("value");
                println(cost);

                return cost;



fonction  getNearestPlace(String start,ArrayList funSearchResults) : return FunSearchResult

                int lastcost = costTime;

                pour (int i=0;i<funSearchResults.size();i++){
                        long cost = calculateTimeCost(start, funSearchResults.get(i).getAddress());
                        si (lastcost>cost){
                                funSearchResult = funSearchResults.get(i);
                                lastcost = cost;
                        }
                }
                print(funSearchResult.getAddress()+funSearchResult.getTitle());
                return funSearchResult;
```